Michael Jarrell

Congruences

1989

version 1997

Documentation d’exploitation

© 1990, 1997, Ircam - Centre Georges-Pompidou

Directeur de la publication : Laurent Bayle

Directeurs délégués : Vincent Puig, Directeur de la Valorisation et Marie-Hélène Serra, Directrice de la Pédagogie.

Textes établis par les départements de Production (Serge Lemouton, Nicolas Vérin) et par le compositeur.

Textes réunis par Marc Battier, responsable de la documentation, Direction de la Valorisation.

Tous droits réservés. Toute reproduction de tout ou partie de ce document, sous quelque forme que ce soit, et par quelque moyen que ce soit, est strictement interdite sans l'autorisation écrite de l'Ircam.

Pour toute représentation de l'œuvre, l'utilisateur doit s'adresser à l'éditeur :

Henri Lemoine, Paris

Deuxième édition, janvier 1998

Réf. bib. :

Michael Jarrell, Congruences (1989), documentation d'exploitation (M. Battier, S. Lemouton, N. Vérin), éditions Ircam - Georges-Pompidou, Paris, 1998.

Ircam
Institut de recherche et de coordination acoustique/musique
1, place Igor-Stravinsky • F-75004 Paris
Tél : 01 44 78 12 33
Fax : 01 44 78 15 40

Contenu

Présentation générale de Congruences (1989)
— avec version 1997 ... 1

Liste des éléments d'exploitation ... 5

Liste de l'équipement électroacoustique ... 7

Schéma de diffusion ... 9

Préparation de l'environnement Midi ... 11

Préparation de l'exécution ... 13

- Max ... 13
- Utilisation des synthétiseurs ... 14
- Préparation des synthétiseurs ... 14
- Chargement des synthetiseurs DX7 et TX802 ... 14
- Préparation de la NeXT ... 15
- Tests ... 17
  - Tests des entrées audio ... 17
- Tests Midi et suiveur ... 19

Patch Max principal pour le Macintosh ... 21

Schéma de diffusion - 1989 ... 25

Eléments d'analyse technique, par Nicolas Vérin ... 27

- Description de la pièce et du dispositif ... 27
  - Remarques sur le schéma MIDI: ... 27
- Historique ... 29
- Description du programme ... 29
- Contrôles MIDI ... 32
- exemple 1 ... 33
- exemple 1.bis ... 34
- exemple 2 ... 35
- exemple 3 ... 36
- exemple 4 (section 2, événement 8) ... 37
- Commandes ... 37
- exemple 5a ... 38
- exemple 5b ... 39
- exemple 5c [suivS1] ... 41
- exemple 5d ... 42

Mesure du tempo 42
exemple 6 43
exemple 7 (section 1, événements 15 à 17) 44
exemple 8 (section 6, événements 3 à 5) 45
exemple 9 (section 2, événement 8) 47
exemple 10 (section 3, événements 17 bis à 19 bis) 48
exemple 11 (section 9, événement 6) 49
exemple 12 (section 1, événement 6; section 10, événement 1) 50
exemple 13 (section 6, événements 13 à 15) 52
Remarques diverses - Conclusions 53
L'utilisation de la CAO dans Congruences - par Francis Courtot et Michael Jarrell 55
Introduction 55
L'origine du projet 55
Méthodologie et mise en œuvre 59
Catégorisation. 59
Engendrement 60
L'utilisation du programme par Michael Jarrell. 61
Notes du programme de la création 63

Présentation générale de Congruences (1989) — avec version 1997

Commande de l'Ircam. Œuvre réalisée à l'Ircam.

durée totale de l'œuvre: 16 min.

Création: 22 et 23 novembre 1989

Grande salle du Centre Georges Pompidou, Ensemble InterContemporain; direction: Peter Eötvös; Pierre-André Valade, flûte MIDI, Didier Pateau, haubois.

Assistants musicaux: Jan Vandenheede, Nicolas Vérin; version 1997 : Serge Lemouton

Effectif

flûte - MIDI solo
hautbois solo
clarinette si b.
clarinette basse si b
cor en Fa
trompette en Ut
trombone ténor - basse
cloches de vaches

synthétiseur DX7II (aussi piano)
clavier de contrôle MIDI KX88

2 modules de synthèse Yamaha TX802

dispositif électronique temps réel

percussion 1:
- 3 cymbales (petite, moyenne, très grande taille);
- grosse caisse très grave;
- timbale;
- crotale;
- cloches-tubes.

percussion 2:
- 3 cymbales (cloutée, moyenne, grande);
- 4 bongos (aigu, médium);
- marimbaphone;
- cloche-plaque (sinon gong)
- accessible aux deux exécutants:
- 3 tam-tams (médium, grave, très grave);
- vibraphone;

alto
violoncelle
contrebasse

VERSION 1997

Liste des éléments d'exploitation

• 1 CD-Rom contenant tous les éléments d’exploitation de l’œuvre.

Liste de l'équipement électroacoustique

1 station d’informatique musicale de l’Ircam NeXT
— 3 cartes IPSW
— 4 ADC (carte 1, ADC 1 à 4)
— 4 DAC (carte 1, DAC 1 à 4)
— 1 interface MIDI

1 Macintosh
— 1 interface MIDI type Opcode Studio 5
— logiciel Max si nécessaire (version 3.5.x pour 68K)
— logiciel Galaxy (pour le SY99, utilisé en remplacement des disquettes)

- 2 synthétiseurs Yamaha TX802 (identifiés comme TX802 A et TX802 B)

- 1 synthétiseur Yamaha SY99 avec une cartouche d'extension RAM

- 1 synthétiseur Yamaha DX7II

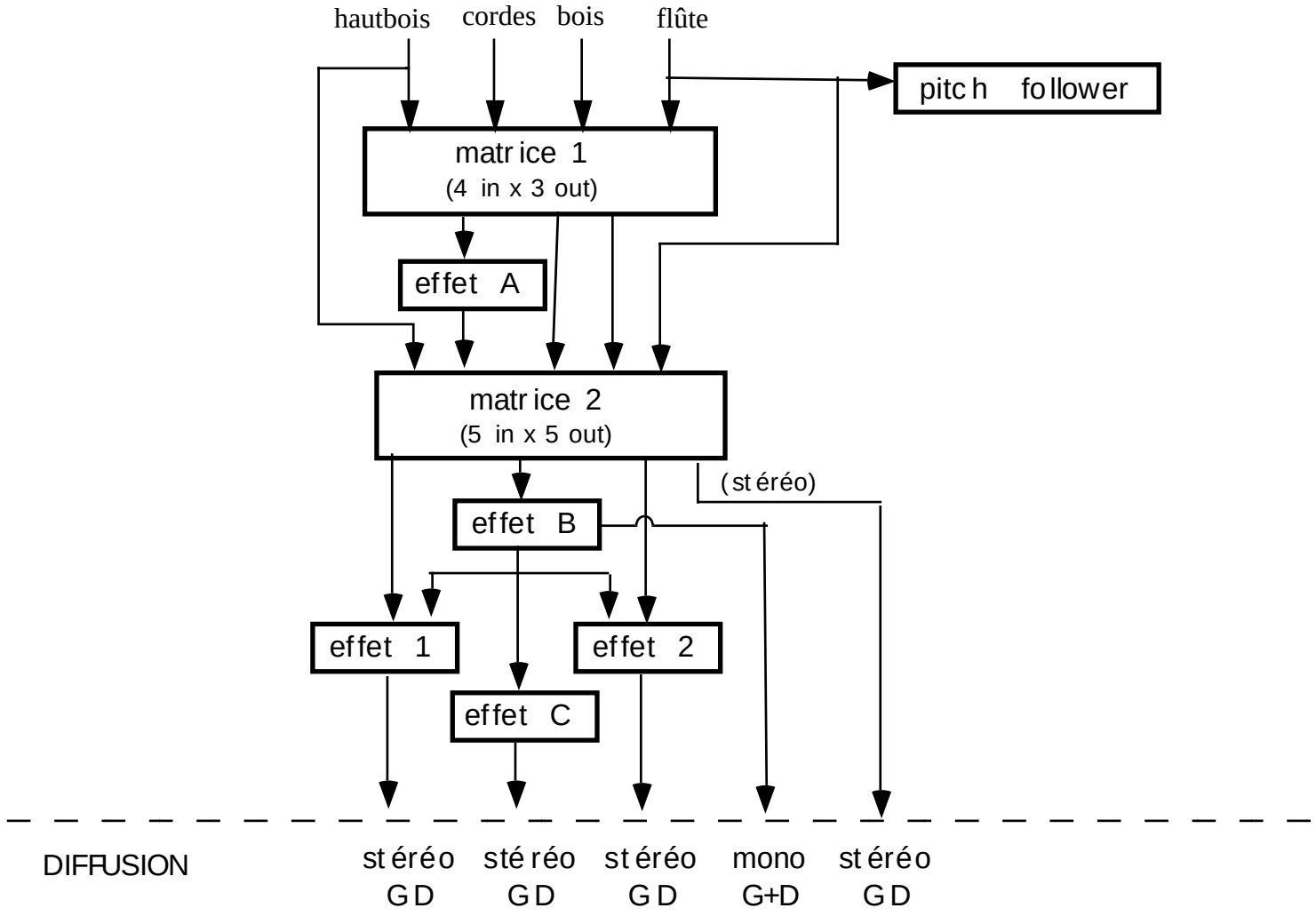
- 5 pédales sustain (switch), réparties comme suit :
    - 2 pédales sustain (switch) branchées sur les entrées FS1 et FS2 du synthétiseur Yamaha SY99
    - 1 pédale Sustain pour le Yamaha DX7I
    - 1 pédale Sustain pour la flûte
    - 1 pédale Sustain pour le hautbois

- 1 dispositif de conversion pédale-Midi type MCS2

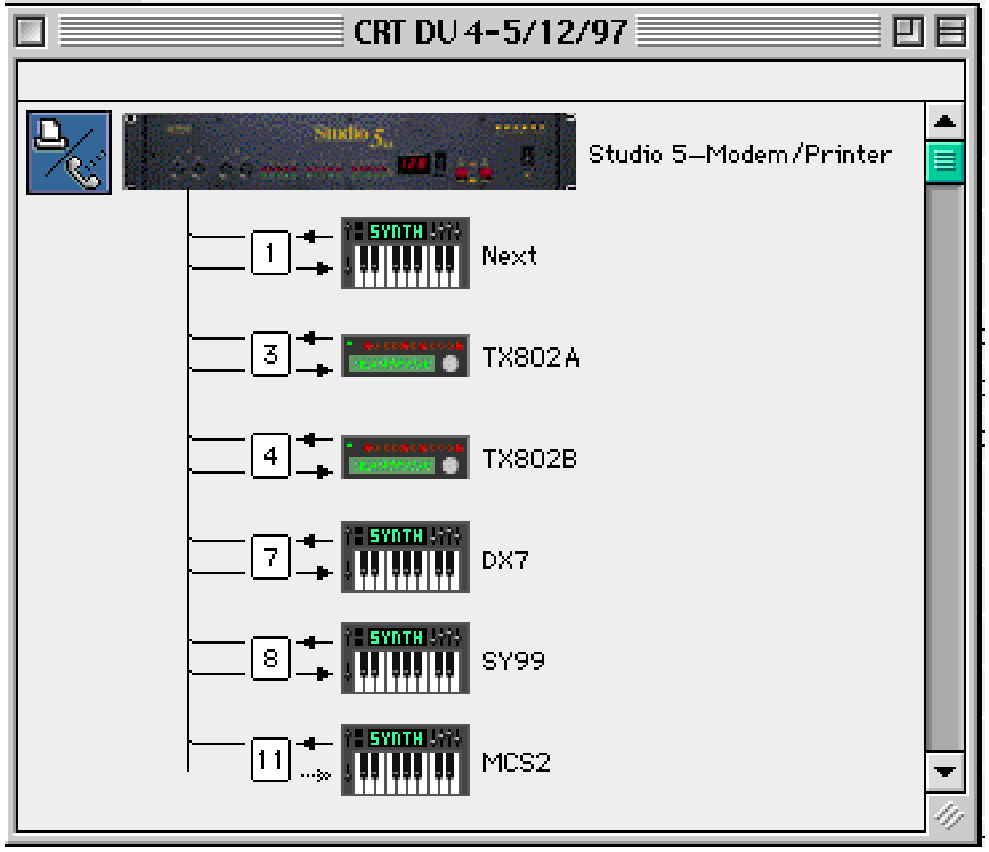
- Microphones pour les instruments

- Micro interne pour la flûte

Schéma de diffusion



Cette figure présente l'état de la configuration OMS (OMS SETUP).



Max

Max

Avant de lancer Max, copier les deux fichiers d'alias « jarrell1 » et « jarrell1 » dans le dossier « aliases » de votre dossier Max. Si le dossier « aliases » n'existe pas, créez-le, puis copiez ces deux fichiers.

Lancez le programme Max.

Ouvrez Options>File Preferences…

Vérifiez que « aliases » figure dans les préferences, sinon rajoutez-le comme illustré dans la figure ci-dessous:

Passer en mode overdrive

Ouvrir le patch « *CONGRUENCES-97-38light.pat ». Vous devez maintenant vous occuper du chargement des synthétiseurs et de la NeXT.

Utilisation des synthétiseurs

DX7 : utilisé comme un instrument; midi non connecté (sauf pour dump). 1 pédale sustain.

TX802 A : changement de programme et volume contrôlés par l'entrée MIDI

séquences en provenance du Macintosh (fichiers MIDI)

utilise les canaux MIDI 5 à 12

sorties I et II vers diffusion

TX802 B : changement de programme et volume contrôlés par l'entrée MIDI

quelques séquences (ev 1.17 et 1.19)

clavier

sorties I et II vers diffusion

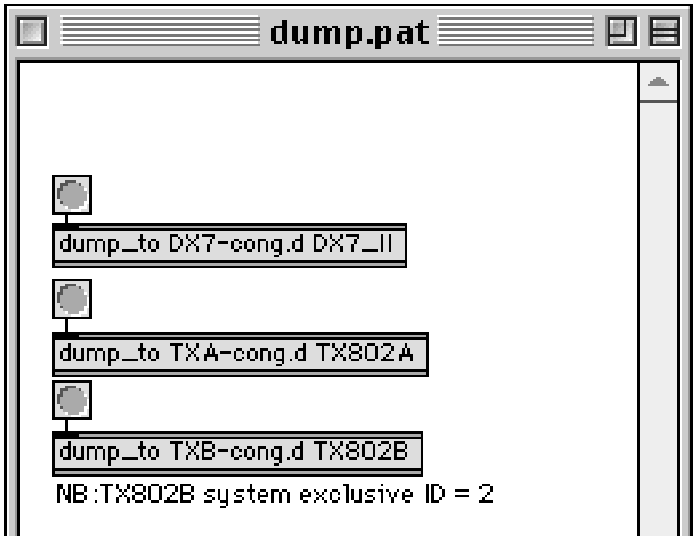
Préparation des synthétiseurs

Ajuster la protection mémoire sur OFF (*Memory Protect Off*) sur tous les synthétiseurs avant d'envoyer les données.

Numéro d'identification de système exclusif : TX 802 (*ID number*) A = 1, B = 2, SY=1, DX7=1.

Chargement des synthetiseurs DX7 et TX802

Ouvrir le dossier Synthétiseurs. Lancez le programme « Congruences-dump.app »



Cliquez sur les boutons correspondants aux trois synthétiseurs.

Vous pouvez aussi utiliser le patch Max « dump.pat » (dans le dossier Synthétiseurs), si vous souhaitez apporter des modifications à ce patch.

Chargement du synthétiseur SY99

Lancez le patch dump.pat (dans le dossier Synthétiseurs)

permet d'envoyer aux synthetiseurs utilises les sons et les programmes necessaires pour l'execution de Congruences ('bulk-dumps')

Préparation de la NeXT

Lancer Max version 0.26

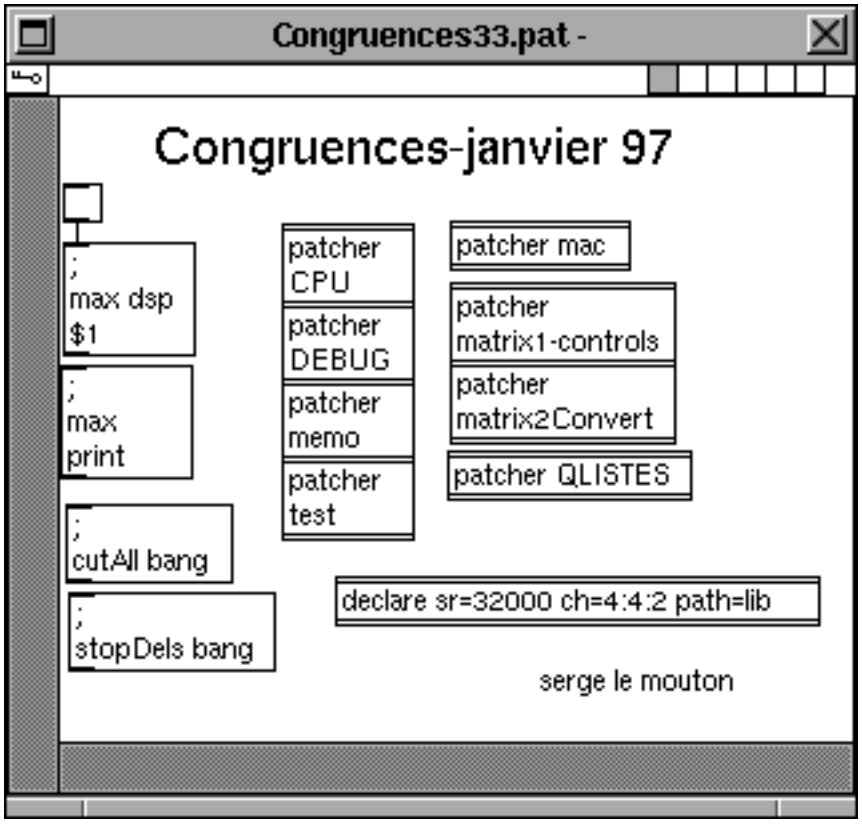
Ouvrir la fenêtre settings, et désactiver les entrées et sorties des cartes 2 et 3.

Ouvrir le patch Congruences35.pat

Ouvrir le patch dtd_dsp.pat

Ouvrir le patch dtd_ctl.pat

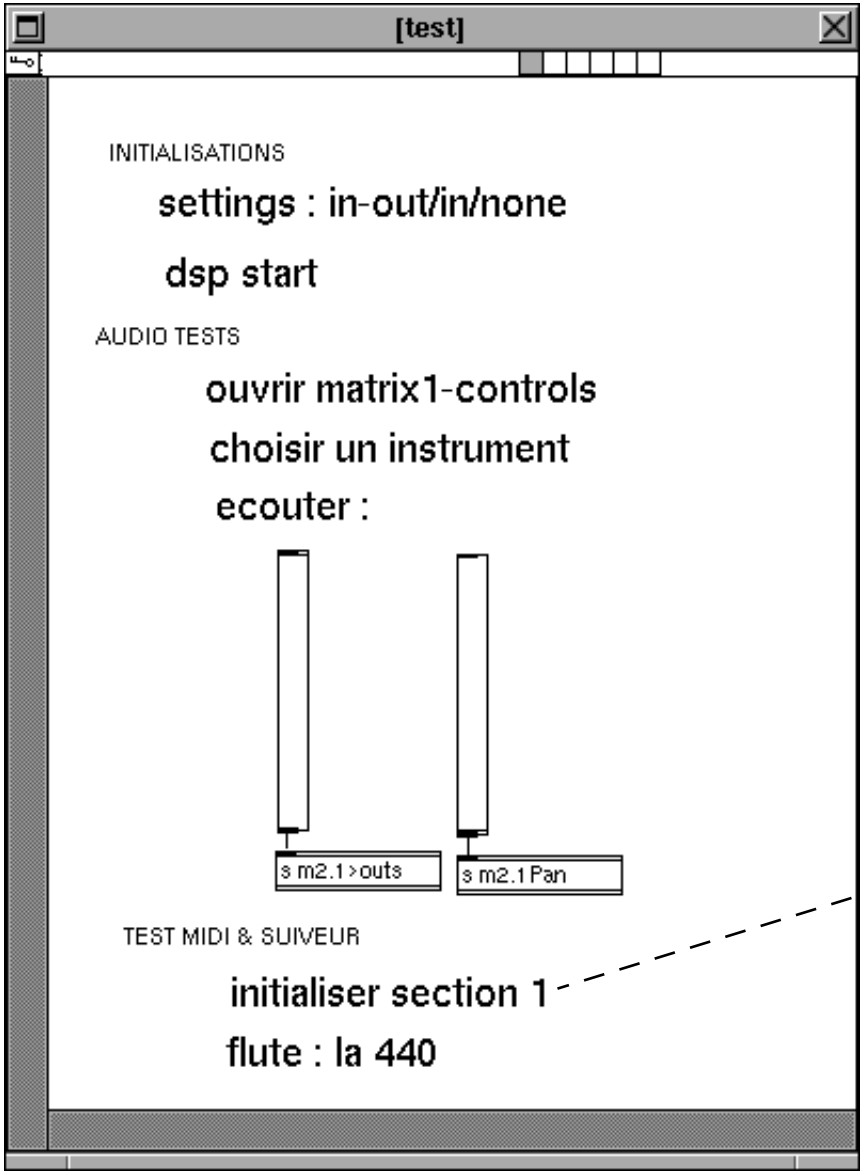
Faire dsp start (toggle sur la fenêtre principale).


Congruences33.pat -
Congruences-janvier 97
;
max dsp
$1
;
max
print
;
cutAll bang
;
stopDels bang
patcher
CPU
patcher
DEBUG
patcher
memo
patcher
test
patcher mac
patcher
matrix1-controls
patcher
matrix2Convert
patcher QLISTES
declare sr=32000 ch=4:4:2 path=lib
serge le mouton

Tests

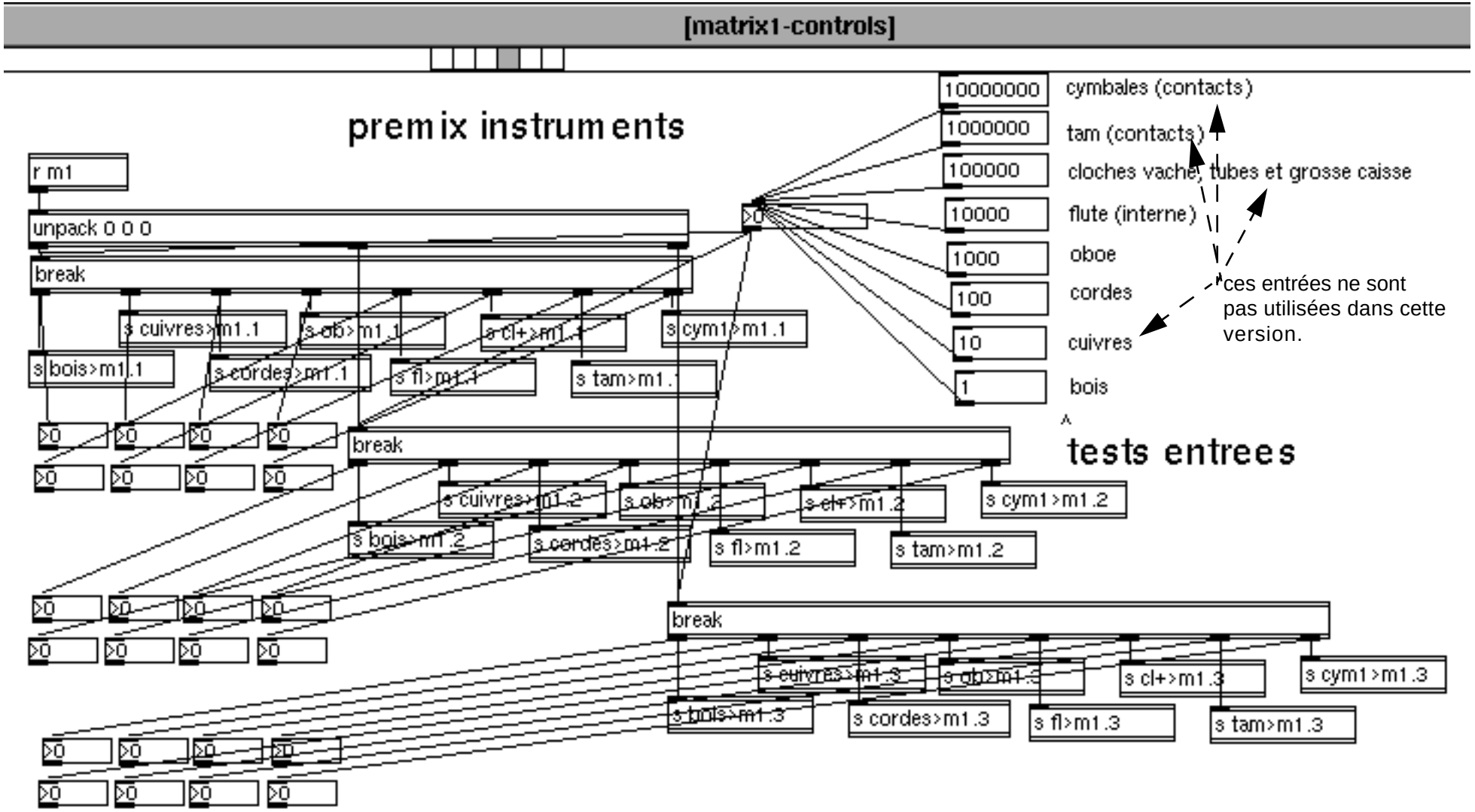
Tests des entrées audio

Ouvrir le patcher test et le patcher matrix1-controls



l'initialisation de la section s'effectue en cliquant sur le bouton «1» du patch Max du Macintosh.

.



.....test-audio-in.tiff.....

Pour chacune des 4 entrées sons (ADC) :

-la sélectionner sur le patch, ce qui a pour effet d'ouvrir la matrice 1 (prémixage des instruments) exclusivement pour cet instrument (ou ce groupe d'instruments).

-Vérifier que le signal provenant du (des) micro(s) concerné(s) traverse bien la machine.

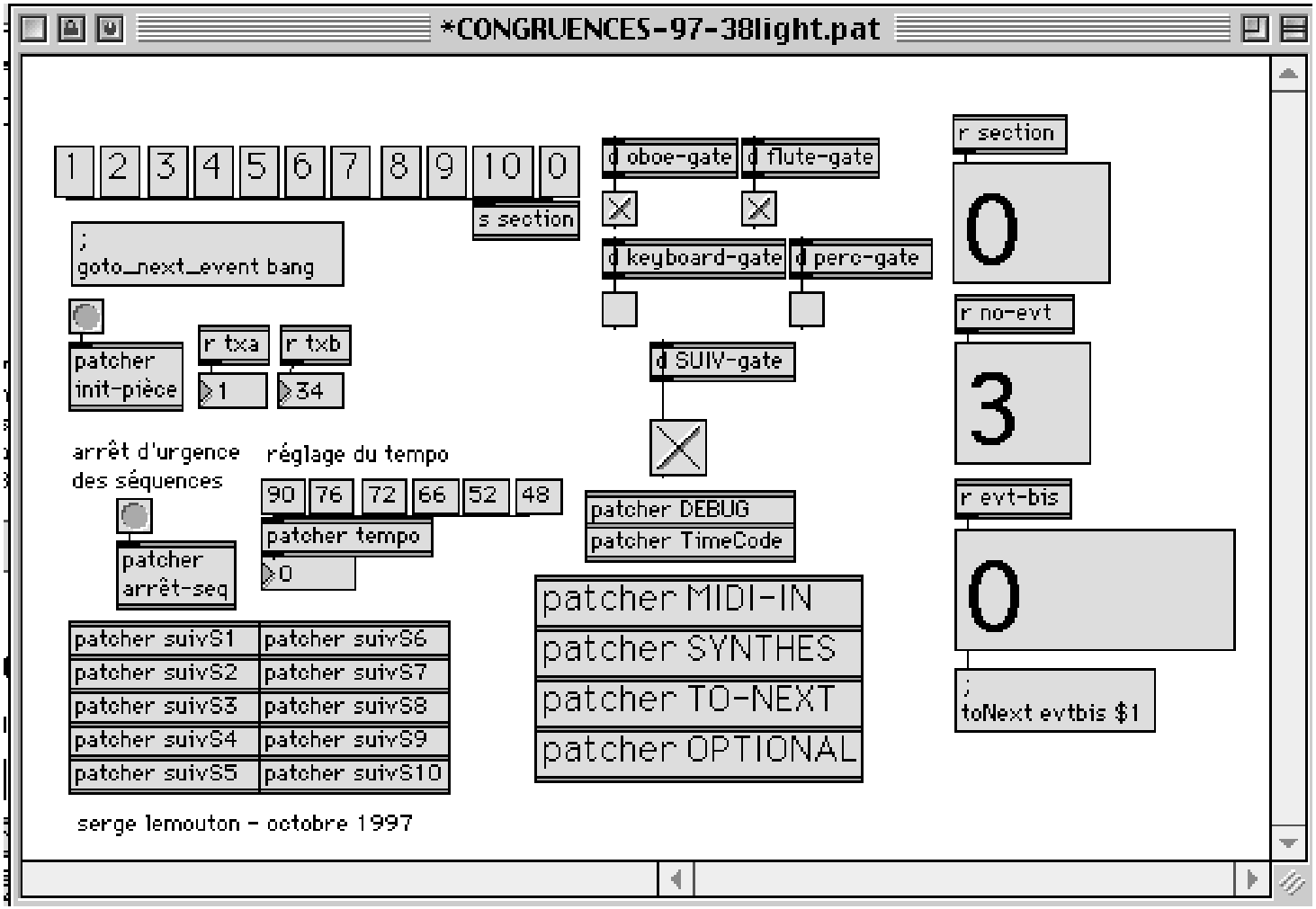
Tests Midi et suiveur

Initialiser la section 1.

Demander au flûtiste de jouer un la 440 (la premiere note de la partition).

Si cette note est figée par la station (réverbération infinie), réinitialiser la section 1. On est alors prêt pour l'éxécution...

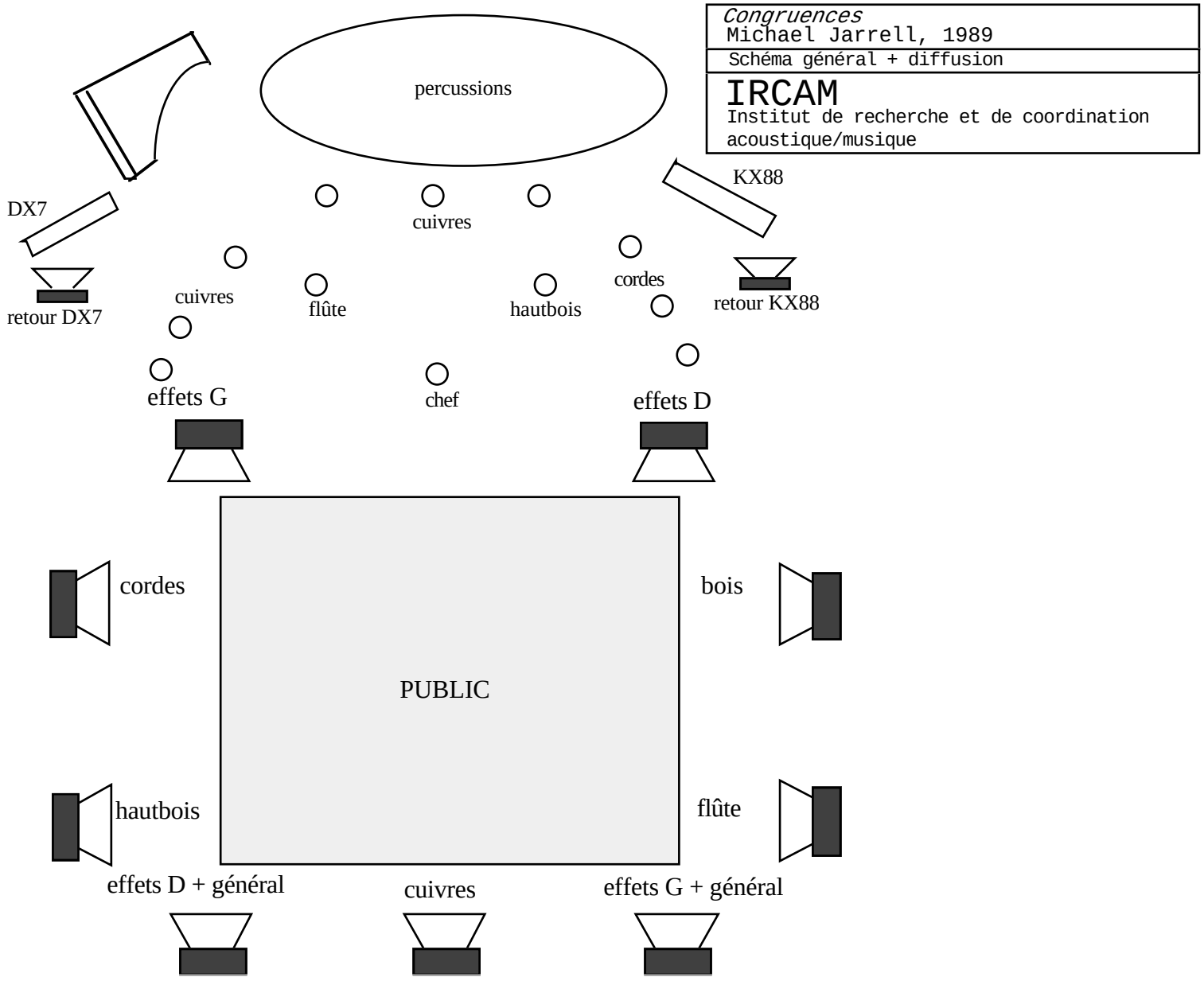
Patch Max principal pour le Macintosh



.

VERSION 1989

Schéma de diffusion - 1989



Description de la pièce et du dispositif

Congruences est une pièce d'une durée d'environ 16 minutes, pour flûte et hautbois solistes, Hautbois.i.Flûte;ensemble et dispositif électro-acoustique contrôlé par ordinateur. L'ensemble est constitué de : clarinette, clarinette basse, basson (également contre-basson), cor, trompette, trombone, deux percussions, deux claviers (dont un jouant piano et DX7 II DX7 IIet l'autre un clavier de contrôle KX88KX88), alto, violoncelle et contrebasse.

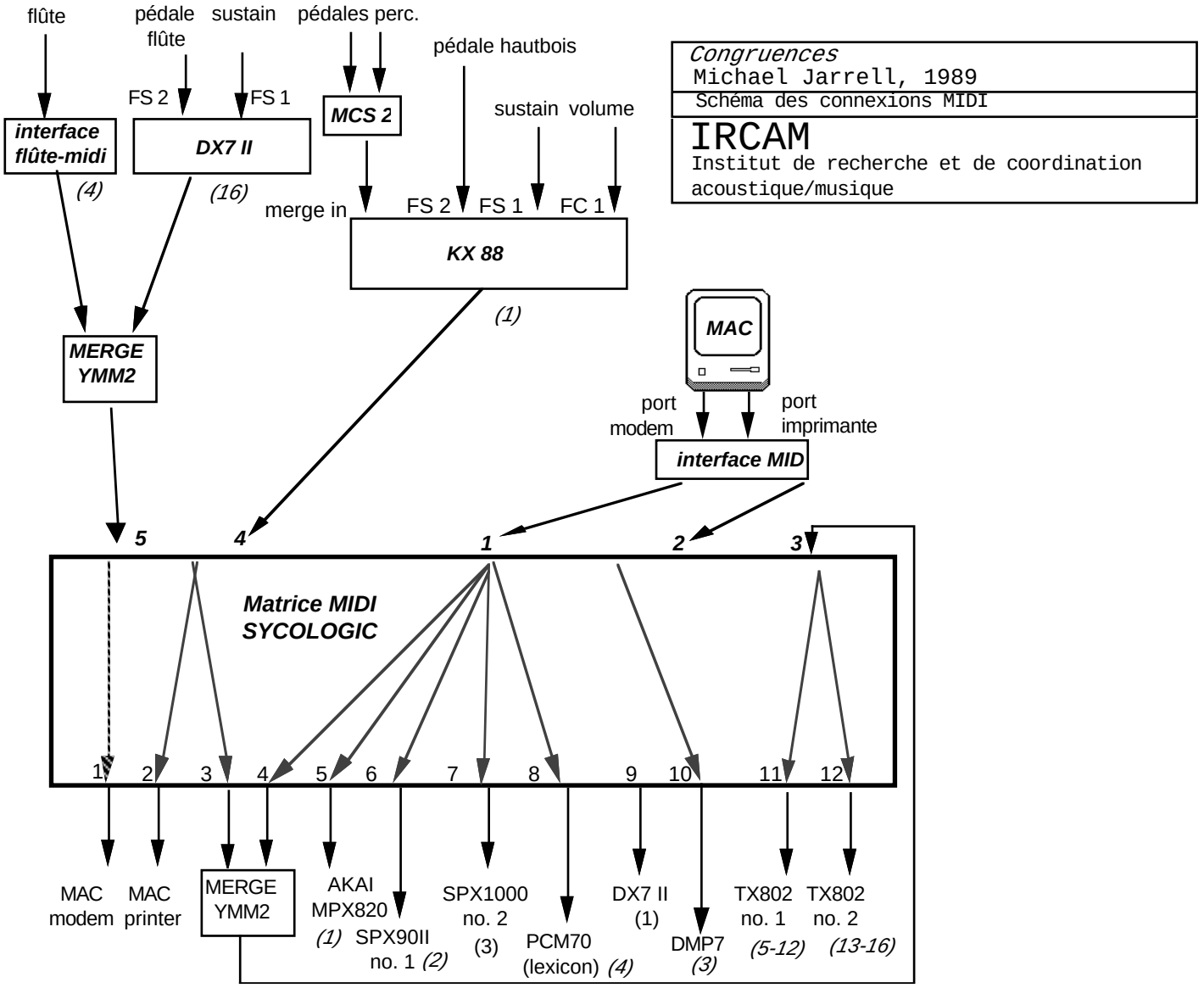
Le compositeur, dont c'est la première expérience avec l'électronique, a cherché à réaliser avec des systèmes personnels, l'équivalent de certains effets obtenus à partir de moyens plus importants.

Le dispositif électronique comporte, outre les deux claviers déjà mentionnés, deux modules de synthèse FM Yamaha TX802, TX802.i.Yamaha;trois processeurs d'effets numériques (un Yamaha SPX1000, SPX1000un SPX90SPX90II, un LexiconLexicon PCM 70PCM 70), deux consoles contrôlables par MIDIMIDI (Yamaha DMP7DMP7 et AkaiAkai MPX820MPX820), ainsi que microphones, microphones de contacts, plusieurs consoles de mélange (pour les pré-mixages vers les effets, ainsi que pour la sonorisation) et amplification en neuf voies.

Bien que n'étant pas idéale vis-à-vis de l'aisance d'utilisation et de la qualité sonore (le signal parcourt un circuit traversant jusqu'à 6 étages), cette configuration permet un maximum de possibilités. La console Akai, dont les possibilités de contrôle par MIDI se limitent à des changements de programmes appelant des "scènes" (c'est-à-dire des états fixes de chacun des paramètres de la console), a été utilisée plutôt comme une matrice d'interconnexion. Elle aiguille les sources entrant dans la SPX1000SPX90II et permet à un ou plusieurs sous-groupes de l'ensemble — cymbales, tam-tams, cordes, etc. — d'accéder à la DMP7, DMP7celle-ci ne disposant que de huit entrées.

Remarques sur le schéma MIDI:

§ l'entrée du DX7 n'est branchée que pour le chargement des programmes de sons.
§ lors des chargements des programmes de sons (TX802 et DX7), la configuration de la matrice MIDI Sycologic doit impérativement éviter les merge.
§ les chiffres entre parenthèses indiquent le canal MIDI de réception ou d'émission.



Congruences
Michael Jarrell, 1989
Schéma des connexions MIDI

IRCAM
Institut de recherche et de coordination acoustique/musique

Historique

La production s'est déroulée sur une période d'un an et demi environ (entre avril 1988 et novembre 1989), mais avec des coupures, portant le total de temps passé à l'Ircam à 10 mois, avec comme tuteur Jan Vandenheede Vandenheede Janpour la première moitié et Vérin NicolasNicolas Vérin pour la suite.

Les premiers essais ont consisté à déterminer la configuration optimale par rapport au résultat désiré, à essayer tous les traitements de sons instrumentaux au moyen des boitiers d'effets, seuls, en série, à partir d'enregistrements mais aussi parfois avec les solistes, ce qui a conduit progressivement à dessiner la structure définitive du programme Max.

Description du programme

Le programme qui a été développé sous Max MAXpour la réalisation de l'œuvre de Michael Jarrell, *Congruences*, a pour but de commander toutes les opérations MIDI - transformations ou synthèse - en suivant les actions des musiciens, de façon à ce que ceux-ci soient libres de se concentrer sur leur propre jeu et ne soient pas esclaves de l'électronique, comme ce serait le cas avec une bande. La première partie de ce programme, appelé `E/S midi`, gère toutes les entrées/sorties. Ensuite vient la partie des `commandes`, qui regroupe le panneau de commande, les suiveurs pour chaque section, et les initialisations de début et de milieu de section. Cette partie intègre également les `sections`, qui reçoivent des suiveurs les numéros d'événements et réalisent les actions correspondantes.

Il est clair qu'une telle partition doit être écrite en gardant constamment à l'esprit , d'une part, les possibilités des machines employées, et d'autre part, quel instrument doit être suivi par le programme et comment (voir le manuel de Philippe Manoury pour les précautions à prendre concernant le suiveur de partition [1] [2] [3] [4] [5]).

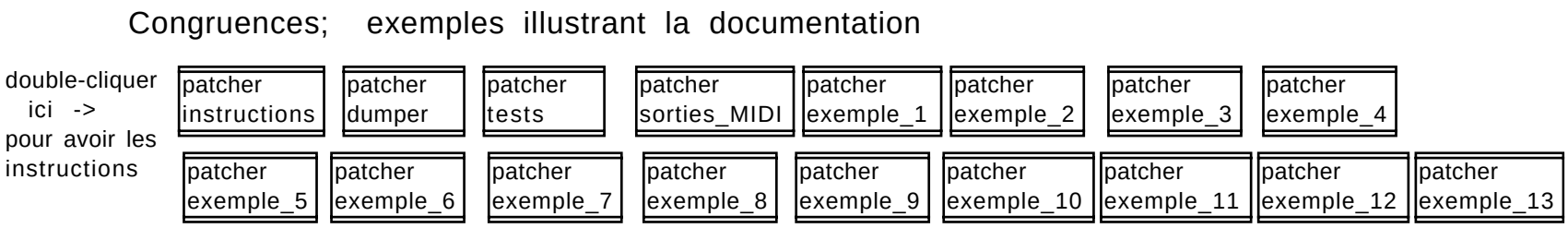
Voici une description détaillée de chaque partie du programme, avec exemples à l'appui. Ceux-ci sont représentés ici de façon graphique, mais il est plus intéressant de les faire fonctionner sur un Macintosh Macintosh; pour pouvoir les essayer avec du son, il convient d'avoir le matériel mentionné au début de cette documentation. Une fois raccordé en suivant le 1er schéma, ouvrier, sous Max 53, le fichier `EXEMPLES`.

---

1. Manoury, Philippe. Max, le manuel d'utilisation. Documentation interne Ircam, Paris, novembre 1988.

2. Si vous désirez sauvegarder les données de la SPX1000, suivez les instructions la concernant, figurant en bas à droite du programme **dumpe**r.

3. voir Valade, Pierre-André. L'interface Flute-Midi (3ème génération) Manuel d'utilisation. Documentation interne Ircam, Paris, mars 1989.



Ce fichier comprend 17 patchers situés au premier niveau. Le patcher `instructions` vous donne une indication de la marche à suivre.

Parcourir les exemples dans l'ordre. La première ligne sert à préparer l'installation nécessaire aux essais sonores en studio. Si vous ne disposez pas du matériel nécessaire (voir document papier), vous pouvez passer directement aux exemples numérotés, qui auront tout de même pour la plupart un effet visible sur l'écran.

---

4. Les conventions suivantes ont été adoptées pour le programme entier (à l'exception des **gat**e des musiciens qui doivent recevoir un 1 ou un 0) :

("?" = caractère quelconque y compris aucun ; "*" = chaîne quelconque de caractères)

a) **dmp7?-*** : l'envoi d'un bang déclenche la baisse puis la remontée du potentiomètre décrit;

b) **dmp7?-*i**n ou **dmp7?-*ou**t : idem avec seulement la montée ou la descente;

c) **dmp7?V*** : requiert l'envoi d'une Valeur, transmise directement au potentiomètre décrit;

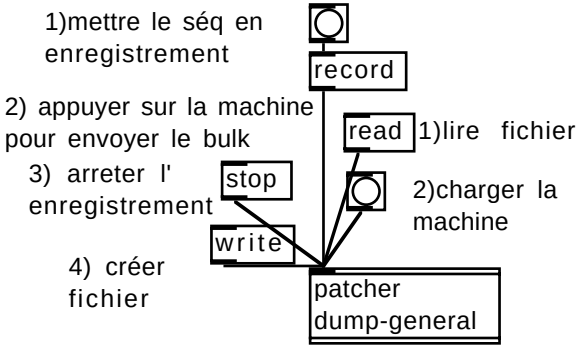
d) **pcmVr**t, **dmp72Vpara**1, … Valeurs de temps de réverb, du paramètre 1 de l'effet 2, etc.

dmp7 correspond au départ principal ; dmp71, dmp72, dmp73 aux départs effets.

5. voir l'exemple 12 ci-dessous.

Il faut ensuite charger les machines à l'aide du programme `dumper`.

A gauche : procédure de sauvegarde A droite : procédure de chargement



Notes pour la sauvegarde :

-dmp7 : il faut envoyer successivement les quatre types de bulk (MEM1-30, PGMC, NOTE, CNTL)
-spx1000 : envoyer un bulkout ALL
-PCM70 : rallumer la machine, puis appuyer sur 7, ensuite LOAD afin d'envoyer le bulk.
-TX802 : envoyer successivement les quatres types de bulk suivants : 1-32 et 33-64 voice data, 1-64 performance data, et system set-up
Attention à ne pas envoyer d'autres informations midi pendant la durée de l'enregistrement, qui dure 50" (sinon elles seront renvoyées lors du chargement), et à effectuer le dump dans ce délai (donc ne pas trop attendre après la mise en enregistrement du séq pour envoyer le bulk).

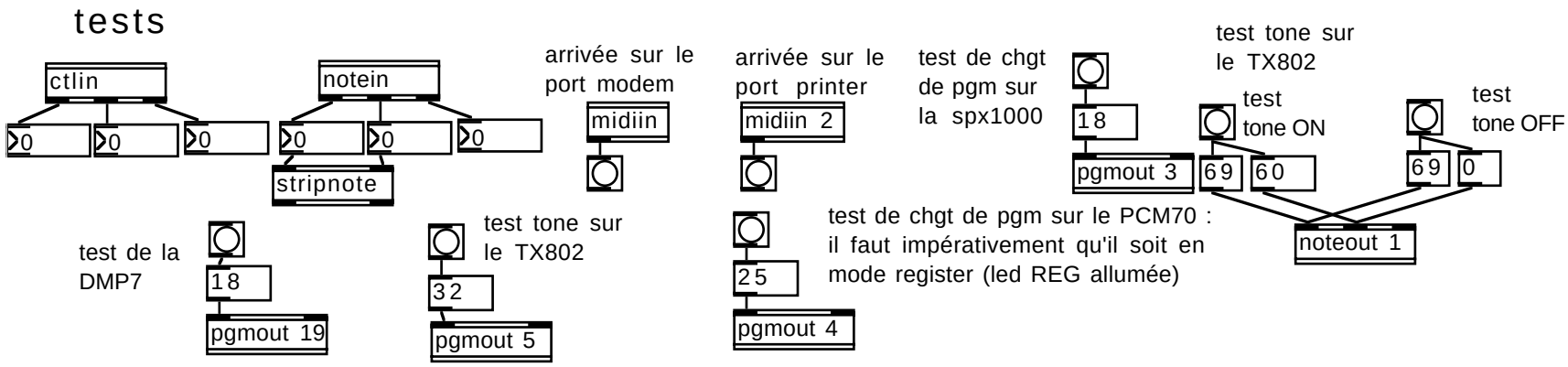
Notes pour le chargement:

Ne pas passer par une mergebox pour charger les machines. Enlever les protections mémoires. Attendre suffisamment (le chargement peut avoir lieu en plusieurs paquets espacés de quelques secondes, et avoir une durée totale d'une minute max.) avant de couper la connection midi. Attention aux TX 802 : le A doit être sur sysex device number 1, le B sur 2

Pour charger la SPX1000, SPX1000appuyer sur "read" puis sélectionner le fichier `SPX1000 définitif`. Vérifier que la sortie modem de l'interface MIDI est bien raccordée sur la SPX1000, dont la protection mémoire doit être enlevée. Assurez-vous que vous avez sauve[6]gardé les programmes de la SPX1000 qui sont éventuellement à conserver , car le chargement va effacer toutes les données qu'elle contenait précédemment, puis appuyer sur le bouton rond, ce qui déclenche la lecture du fichier contenant les codes system exclusives de chargement. Répéter ensuite l'opération pour le PCM70 et la DMP7. Trouvez un son se rapprochant plus ou moins de la flûte sur le synthétiseur ou l'échantillonneur.

6. Ces problèmes devraient être largement résolus par l'emploi du nouveau module de Max, appelé **qlis**t.

Vérifiez les canaux MIDI de toutes les machines. Le patcher `tests` vous aidera à vérifier l'ensemble de la configuration:



Contrôles MIDI

Les instruments que le programme peut suivre sont :

1) la flûte, au moyen de l'IFM (interface flûte-MIDI);

Celle-ci a été spécialement développée pour cette pièce, à partir de l'interface flûte-MIDI existante. MIDI.i.Flûte-MIDI;Au moyen de micro-interrupteurs placés sous les clés de la flûte, l'interface décode les doigtés joués par rapport à une table de correspondances, et transmet des messages MIDI standard donnant les hauteurs jouées (note-on et note off, mais seuls les note-on sont pris en considération par le programme) . Afin de renforcer la fiabilité de certains passages (notamment en cas de notes répétées), et pour pallier à l'impossibilité de prendre comme événement un note-off, nous avons été conduits à ajouter une pédale de type interrupteur pour le flûtiste, qui s'insère dans le circuit en la branchant sur le DX7 IIDX7 Yamaha.

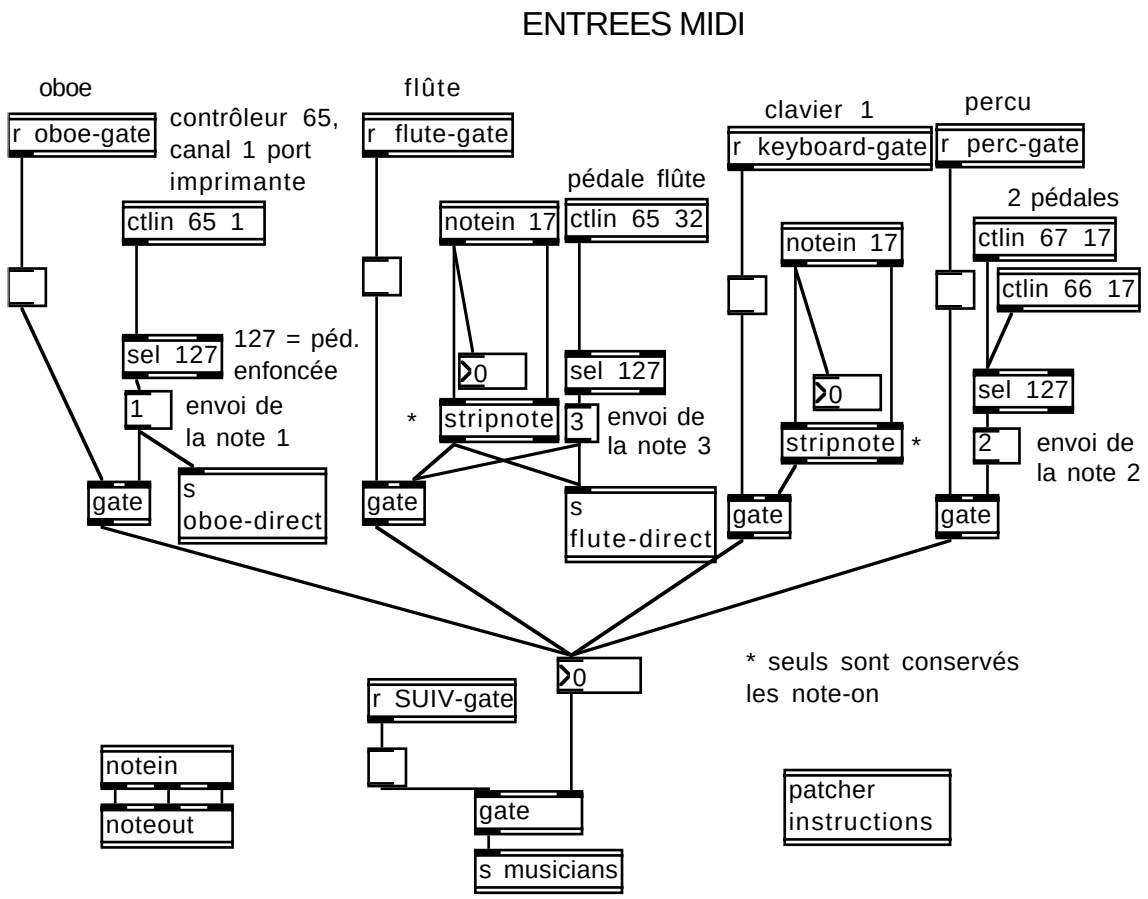
2) le hautbois, Hautboisau moyen d'une pédale de même type, se raccordant au clavier KX88;

3) le clavier KX88;KX88

4) un des deux percussionnistes, au moyen de 2 pédales raccordées à un boîtier MCS2 (les deux pédales sont équivalentes, la deuxième ne servant que pour des raisons de commodité).

Suivant les endroits de la partition, un ou plusieurs instruments rentrent dans le suiveur. Le patcher `musiciens`, qui figure dans l'exemple 1, montre comment cela est organisé.

exemple 1


ENTREES MIDI
oboe
r oboe-gate
contrôleur 65, canal 1 port imprimante
ctlin 65 1
sel 127
127 = péd. enfoncée
1
envoi de la note 1
gate
s oboe-direct
flûte
r flute-gate
notein 17
pédale flûte
ctlin 65 32
>0
sel 127
*
stripnote
3
envoi de la note 3
gate
s flute-direct
clavier 1
r keyboard-gate
notein 17
>0
stripnote *
gate
percu
r perc-gate
2 pédales
ctlin 67 17
ctlin 66 17
sel 127
2
envoi de la note 2
gate
>0
* seuls sont conservés les note-on
r SUIV-gate
gate
s musicians
notein
noteout
patcher instructions

Là aussi, le patcher instructions vous aide dans la marche à suivre:

---

exemple 1.bis

Dans la configuration des exemples qui suivent, le clavier dont vous disposez va remplacer la flûte. Utilisez le patcher test pour vérifier sur quel canal midi sort le clavier, puis ajustez sur le patcher le canal MIDI de la flûte (notein ...). Vérifiez également la pédale, qui doit être branchée sur la prise FS2, qui doit correspondre avec ce qui figure dans la boîte ctlin du hautbois (ctlin 65 17 quand vous ouvrez le patcher). Exemple : vous ouvrez le patcher test et jouez une note sur le clavier. Vous constatez que vous recevez un notein sur le canal 1. Il faut donc modifier le

notein 4 de la flûte en notein 1

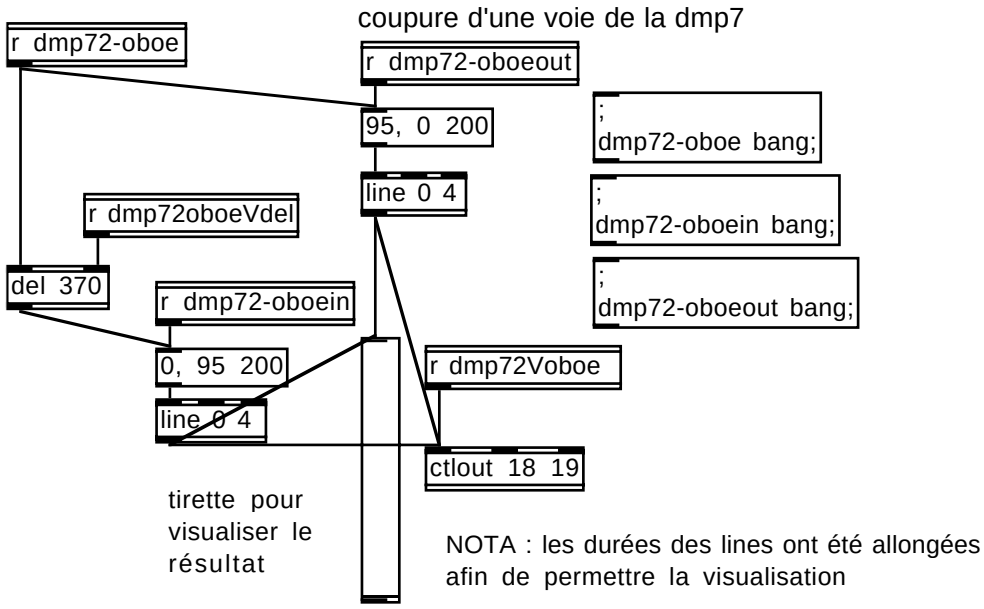
Vous appuyez ensuite sur la pédale et obtenez du ctlin les valeurs suivantes : 127 puis 0, 65, 1 c'.à.d. contrôleur 65 sur canal 1. Il suffit maintenant de modifier la boîte

ctlin 65 17 en ctlin 65 1

Il faut maintenant vérifier la correspondance des canaux MIDI entre le programme et les machines. Réglez d'abord celles-ci de la façon suivante : DMP7 sur canal 3 (le faire sur chaque banque, ainsi que pour la SPX) en sortie du port imprimante de l'interface du macintosh;
SPX sur canal 3, mais en sortie du port modem du mac;
PCM70 sur canal 4. Essayez ensuite les changements de programme au moyen du patcher test. Si tout ne fonctionne pas, vérifier le patch midi.

La variable `musicians` reçoit celui ou ceux des instruments dont le *gate* est ouvert, cette ouverture ou fermeture étant commandée par l'envoi d'un 0 ou d'un 1, respectivement vers `flute-gate`, `oboe-gate`, etc. On peut voir également la possibilité, qui sera discutée plus bas, de recevoir flûte et hautbois en direct, c'est-à-dire quelque soit la position de leur *gate*.
Peu avant le concert, des problèmes de fiabilité de l'interface flûte-MIDI ont entraîné l'ajout d'un *gate* général permettant de couper complètement la réception de notes MIDI, tout en laissant le suiveur en place. Ceci permet (en principe) de rattraper des situations où le suiveur est en avance.
Les autres *patchers* servent tous à contrôler les appareils MIDI. La plupart concernent DMP7la DMP7, qui reçoit des valeurs de niveau pour tous les potentiomètres - entrées, départs effets - mais aussi de type d'effet, de paramètres d'effets, etc. Dans l'exemple 2, c'est le niveau du hautbois dans le départ 2 (2ème effet interne de la DMP7) qui est contrôlé.

exemple 2



Un des problèmes importants qui s'est posé est le bruit occasionné par une unité SPX SPXsi l'on effectue un changement de programme pendant que l'entrée reçoit du signal. Les changements de programmes étant nécessairement très nombreux dans cette œuvre, une façon automatique de contourner le problème a été cherchée. Il suffit d'envoyer un `bang` à l'objet `dmp72-oboe` pour avoir un shunt très rapide (en 35 ms), suivi après 370 ms de délai d'une remontée très rapide. Ces valeurs ont été trouvées après essais co[7] [8]mme étant les plus courtes pour effectuer un changement de programme sans click sur une SPX. On peut aussi fournir SPXune valeur unique qui sera transmise par l'objet `dmp72Voboe` (avec un V comme valeur), ou encore envoyer un bang dans `dmp72-oboein` ou `dmp72-oboeout`, ce qui aura pour effet respectivement de faire baisser ou remonter le potentiomètre. Ceci est reproduit de façon identique pour tous les potentiomètres de la DMP7. DMP7

La console Akai MPX820 MPX820ne reçoit, elle, que des changements de programme. Pour unifier, on ajoute 1 de façon à obtenir sur chaque appareil le programme de même numéro que celui que l'on envoie. De plus, l'Akai n'accepte le changement de programme que si celui-ci est accompagné d'un octet quelconque. L'exemple 3 montre aussi comment les deux SPX reçoivent leur changement de rapport de transposition quand elles sont sur un pro-

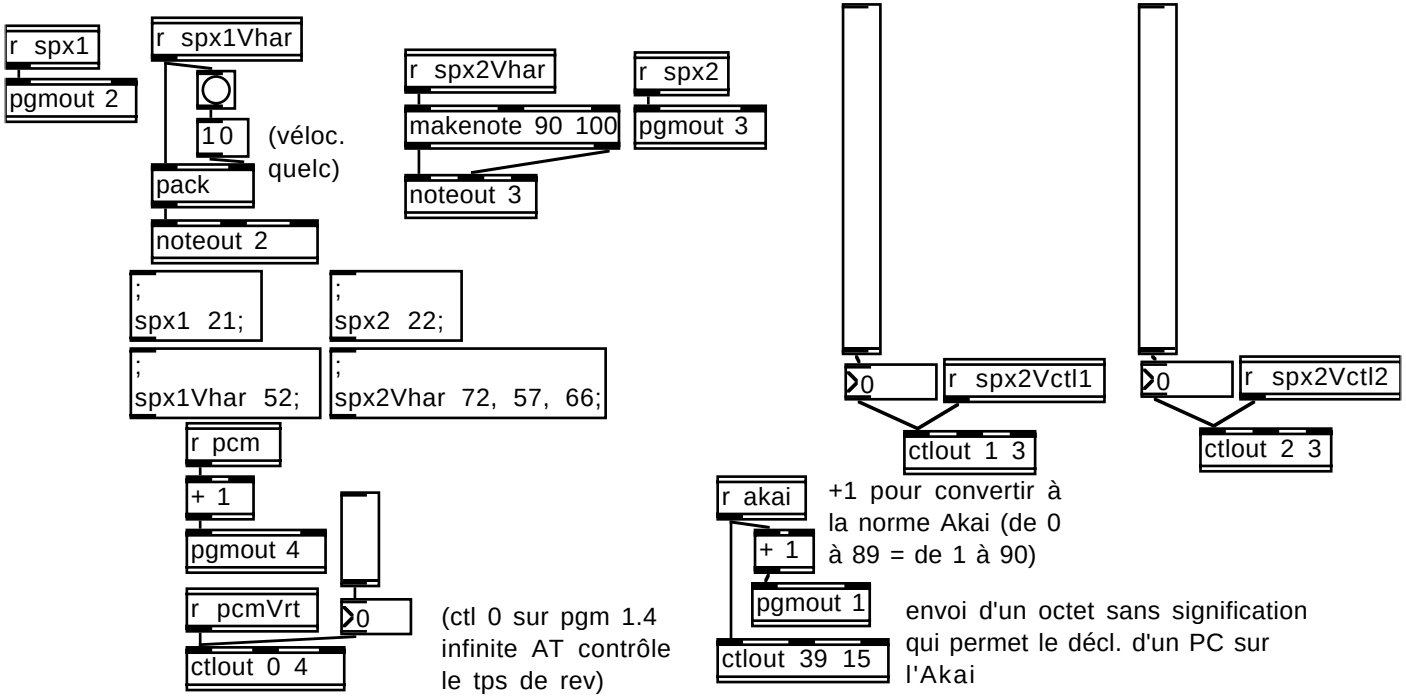
7. Manoury, Philippe. op. cit.

8. à la date de publication, et au format 0 uniquement.

gramme d'harmoniseur.

---

exemple 3



Noter que cela ne fonctionne pas exactement de la même façon pour la SPX90II et pour la SPX1000 (la spx1 est la 90 II, la spx2 la 1000). Cette dernière peut recevoir jusqu'à trois notes simultanément pour modifier les rapports de transposition dans le programme d'harmoniseur triple . La SPX1000 peut également recevoir des contrôles, s'appliquant aux paramètres choisis dans le programme utilisé. Le processeur PCM70d'effets Lexicon PCM70 reçoit, en plus des changements de programme (peu fréquents, car il est surtout utilisé en `infinite AT`), un contrôle du temps de réverbération, allant du minimum (0,23 s) au maximum (infini) pour une valeur MIDI de 32.

Quand aux deux synthétiseurs TX802, TX802ceux-ci ne reçoivent du Macintosh Macintoshque des changements de programme et des notes MIDI. Mais lorsqu'ils sont joués par le clavier (voir le merge dans le schéma MIDI), ils reçoivent alors tous les types de signaux MIDI. Il y a une exception à cela, c'est l'envoi, dans des cas particuliers, de **system exclusive** pour contrôler indépendamment le niveau d'un générateur de l'un des TX. Ceci est montré dans l'exemple 4.

exemple 4 (section 2, événement 8)

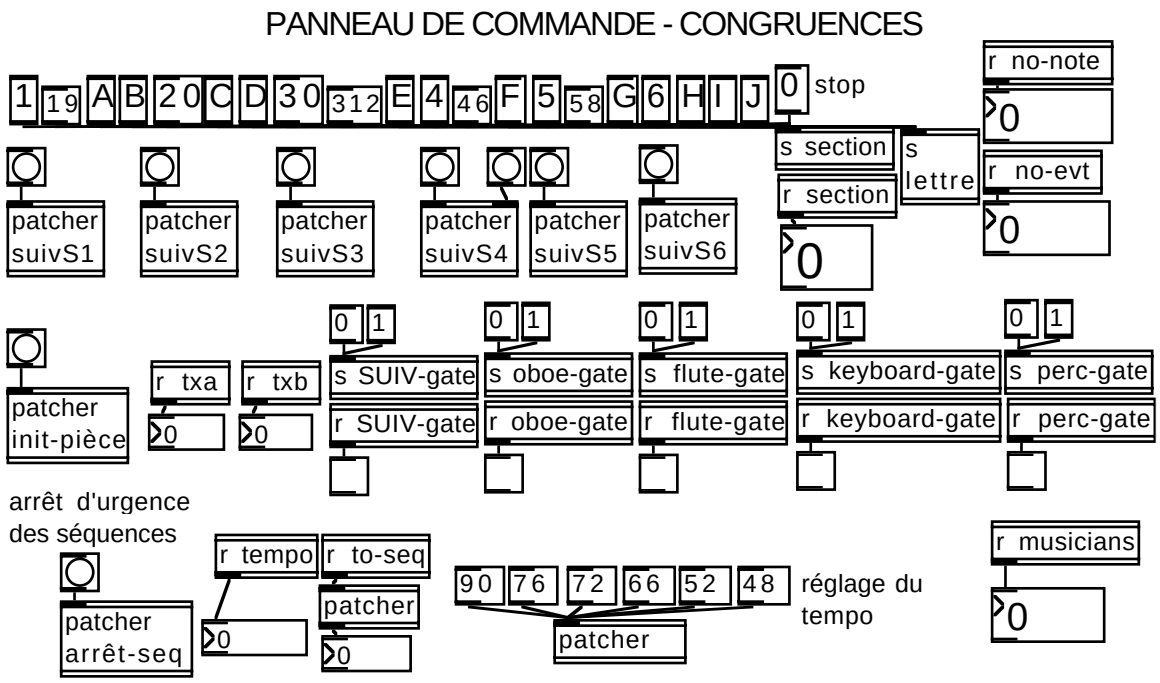
127, 0 1200
line 90 30
240, 67, 16, 26, 32, $1, 247
midiout
240, 67, 16, 26, 32, 127, 247
sysex pour shunter le volume du générateur 1 du TX802
remise au niveau max

On envoie dans l'événement qui précède un unique message **sysex**, qui assure, en cas de retour en arrière pendant les répétitions, la remise au niveau nominal du générateur concerné. Dans le déroulement normal de la pièce, celui reprend son niveau mémorisé au moment du changement de programme suivant.

Commandes

a) panneau de commande

exemple 5a



Tout en haut de l'exemple 5a figurent les numéros de section, de 1 à 10 (écourté dans l'exemple) ainsi que de sous-sections (19, 106...), des lettres correspondant aux signes de répétition de la partition, et enfin un bouton d'arrêt affecté du chiffre 0. Ces numéros servent à positionner le suiveur, ainsi que l'ensemble des machines, soit au début d'une section, soit au milieu. Dans ce dernier cas, le deuxième chiffre indique le numéro de l'événement de la section considérée à partir duquel on s'apprête à repartir (par exemple, le chiffre 19 permet de repartir sur l'événement 9 de la section 1). Leur unique fonction est de gagner du temps pendant les répétitions. Les lettres ont exactement le même rôle que les sous-sections, mais ont été ajoutées peu de temps avant la fin.

Les boutons-poussoirs figurant juste au-dessous servent à déclencher l'événement suivant l'événement courant. Du fait de l'utilisation fréquente de `del`, il est parfois dangereux d'appuyer sur ces boutons plusieurs fois de suite sans attendre. On peut avoir, par exemple, un événement qui provoque un changement de programme sur une machine après un délai de 5 secondes, suivi d'un autre changement de programme sur l'événement suivant. Si on les enchaîne trop vite, le deuxième changement peut précéder le premier, et l'on se trouvera ainsi décalé. Certaines mesures (stops) ont été prises pour éviter ce problème, mais elles peuvent ne pas être totalement efficaces . Aussi est-il prudent de manoeuvrer en observant la fenêtre MAXMAX, dans laquelle s'affichent des messages du type `EV4: dél 4s`, signifiant qu'il faut attendre 4 secondes avant d'envoyer l'événement suivant. Pour rendre les manipulations plus aisées en répétition et en concert, deux dispositifs ont été ajoutés afin de permettre depuis le clavier :

1) d'envoyer l'événement suivant (au moyen de la touche ENTER),

2) d'avancer le Suiveursuiveur d'une note (au moyen de la touche +).

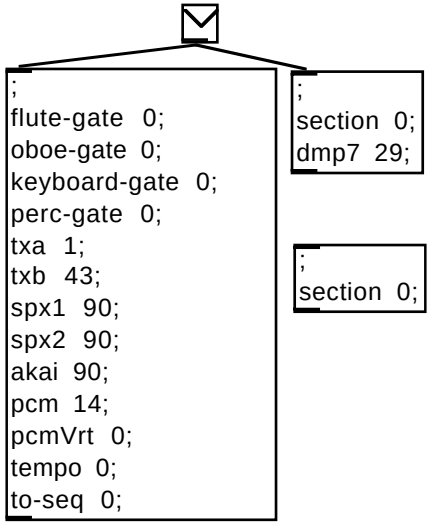
D'autre part, il est possible d'intervenir sur plusieurs choses depuis le panneau de commande. Un bouton d'arrêt d'"urgence" permet de stopper rapidement toute séquence en cours, où que l'on se trouve ; ceci peut être utile en répétition, notamment lors d'un arrêt dans un passage comportant une séquence. Un réglage du tempo des séquences est possible, pour les reprises en répétition directement sur un départ de séquence (c'est-à-dire sans que la mesure du tempo réellement joué ait pu être faite), voire pour corriger un tempo erroné. Ce dernier est visualisé de deux manières : 1) tel qu'il a été mesuré lors de la dernière occasion ("tempo"); 2) tel qu'il est envoyé au séquenceur ("to-seq"), à savoir éventuellement corrigé en fonction d'une moyenne faite avec une mesure précédente, ou encore borné par des valeurs maximales et minimales de sécurité.

Sont visualisés également, les *gates* de chaque instrument suivi, ainsi que le *gate* général du suiveur. Les boîtes 0 et 1 permettent d'intervenir sur chaque *gate*.

La nécessité d'un patcher `init-pièce` s'est fait sentir pendant les répétitions.

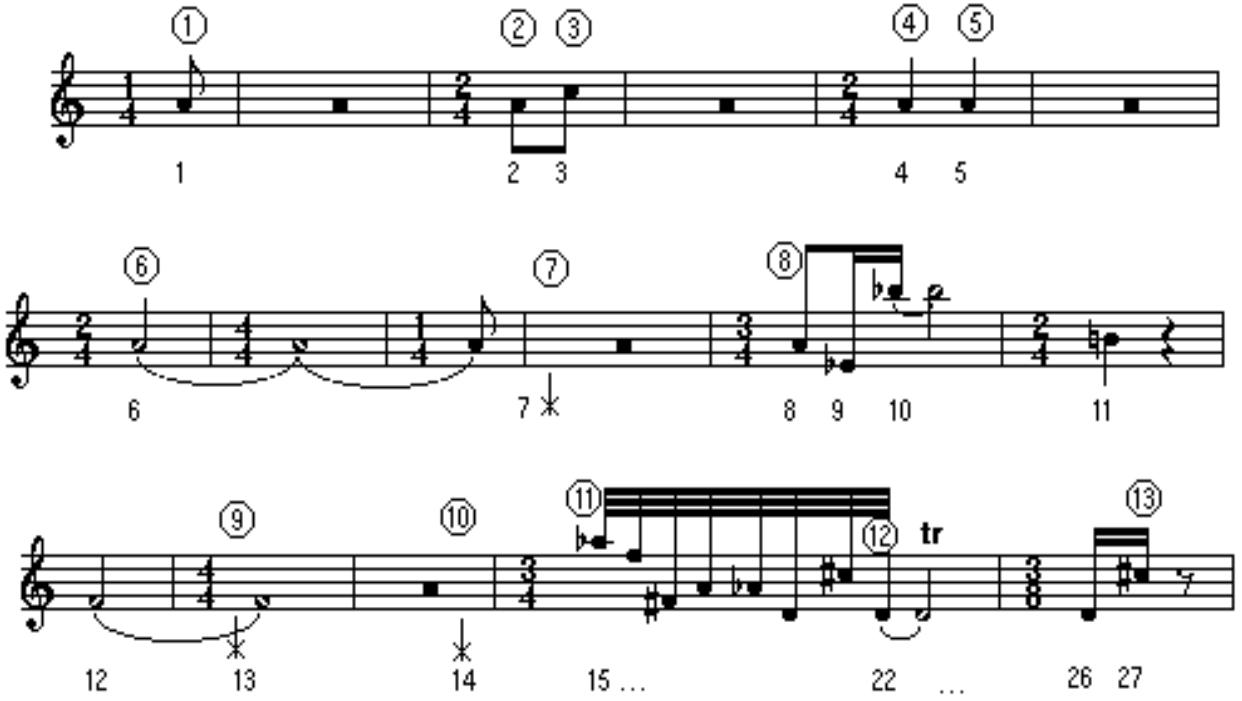
---

exemple 5b



Son but est d'arrêter le suiveur, de fournir au joueur de clavier KX88 un son sur lequel il puisse régler son jeu et déterminer le niveau sonore des retours, et d'initialiser certaines valeurs. Les numéros de programmes envoyés au TX802 TX802sont également affichés sur le panneau de contrôle pour vérification. Cela est très utile, car beaucoup de problèmes peuvent se poser avec ces machines : la moindre défaillance dans la transmission MIDI lors des changements des sons (câble trop long, passage à travers une mergebox, etc.) est suffisante pour causer des patches aléatoires. D'autre part, les changements de sons étant déclenchés par le programme Max, si le suiveur est perdu, le joueur de KX88KX88 n'aura plus les sons correspondant à ce qui est écrit. La fréquence, dans les premières répétitions, de ce dernier problème nous a d'ailleurs amené à laisser le joueur de DX7DX7 faire lui-même ses changements de programmes, car il y a largement le temps et ce synthétiseur est parfaitement autonome dans le dispositif de la pièce.

Les premières transformations concernent ici la flûte. FlûteTout d'abord, le la du début est gelé dans la réverbération infinie du Lexicon PCM70. Puis d'autres notes apparaissent, soit jouées par la flûte, soit transposées de celle-ci par un programme d'harmoniseur sur la SPXSPX90 ou 1000 ; ces autres notes accumulées sont également gelées. A l'événement 6, le la de la flûte stoppe l'accord gelé et provoque une nouvelle transposition, de triton cette fois, qui est arrêtée par l'événement suivant, un coup de pédale du hautbois ; par contre le la gelé dans le PCM70PCM70, lui, reste encore jusqu'à l'événement 8. En 9, le hautbois, eHautboist les vents de l'ensemble parcourent les deux harmoniseurs l'un après l'autre, chacun avec une réinjection qui provoque une cascade. Chaque son est transposé d'une tierce mineure descendante, puis le résultat est lui-même transposé après un court délai, et ainsi de suite, avec pour effet des arpèges de septièmes diminuées appliqués à toutes les notes des vents et du hautbois. Les événements 10 et 11 servent à geler dans le PCM le son résultant du mélange de tous les instruments de l'ensemble, son qui est arrêté dans l'événement 13 qui conclut cet extrait.
Vous pouvez essayer vous-même ce passage dans le studio, en jouant au clavier la partition ci-dessous ; il s'agit de la ligne de flûte simplifiée afin de garder l'essentiel. Noter que les transformations, elles aussi, ont du être simplifiées, et que celles qui concernent d'autres instruments ne peuvent avoir d'effet. Avant de commencer à jouer, appuyer sur le bouton 1 du panneau de commande, de façon à initialiser les machines et à mettre le suiveur en attente. Vous pouvez reprendre à n'importe quel moment en réappuyant sur 1. Le Suiveursuiveur de Max n'étant pas dépendant du tempo, il n'est pas important que tempo et rythme soient exacts.



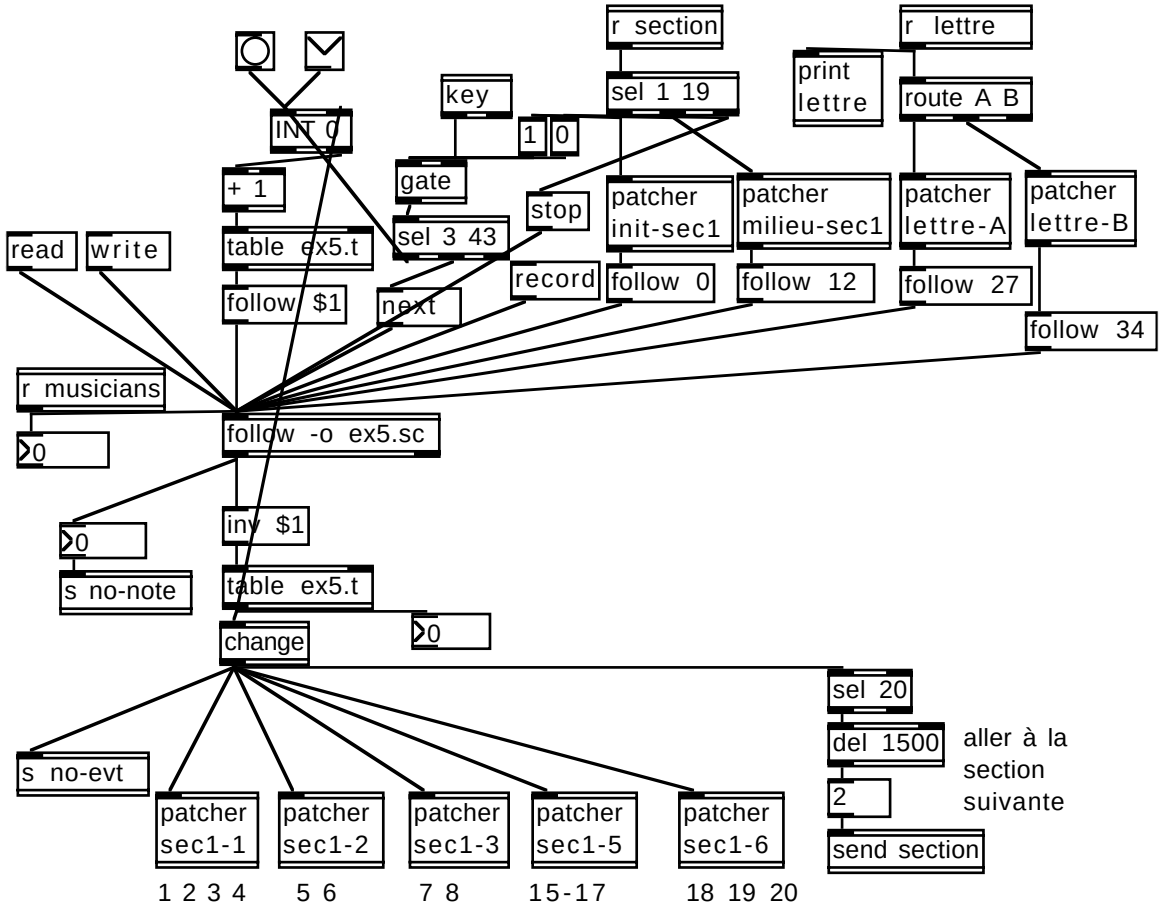
b) suiveurs

Toujours dans l'exemple 5 sous Max, on peut ouvrir le suiveur de la section 1. (patcher `suivS1`; illustré ci-dessous dans l'exemple 5c). Le principe est exactement tel que décrit dans le manuel de Philippe Manoury . Manoury PhilippeIl s'y ajoute un mécanisme de déclenchement des événements pas à

pas, quelque soit le nombre de notes qui les sépare dans la partition (actionné par le bouton du haut). Le même effet est obtenu en appuyant sur la touche **ENTER** ; la touche +, elle sert à avancer le suiveur d'une note (par l'envoi du message `next`).

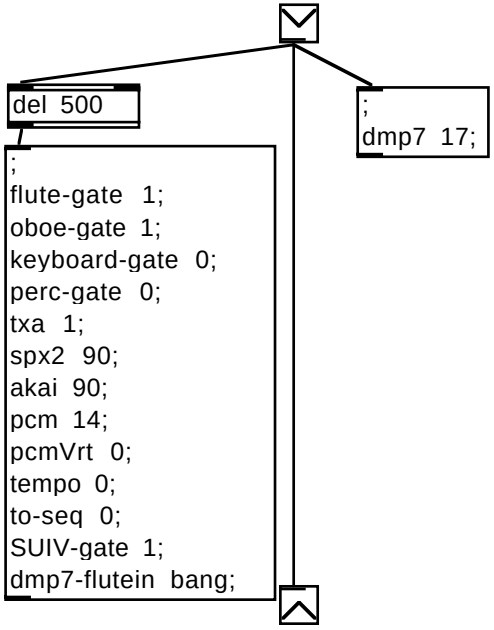
---

exemple 5c [suivS1]



A l'intérieur de cette fenêtre patcher, figure un autre sous-patcher (exemple 5d) intitulé `init-sec1`. Chaque section débute par l'envoi des changements de programmes appropriés à chaque machine, ainsi que le positionnement correct des *gates*. Ces `init-sec` permettent de reprendre à n'importe quel début de section. Toutefois, certains enchaînements (début des sections 2, 3 et 9) ont posé un problème particulier : si un son résonne encore dans une réverbération de la DMP7 DMP7au moment où celle-ci reçoit un changement de programme, sa mémoire se videra, provoquant dans le meilleur des cas un arrêt brutal de la résonance, mais bien souvent assaisonné de clics. La solution apportée consiste à dissocier le cas du déroulement normal de la pièce — où l'on remplace le changement de programme de la DMP7 par les quelques control change nécessaires — du cas de reprise en répétition — où cela ne pose pas de problème de procéder, comme dans les autres débuts de sections, par un changement de programme.

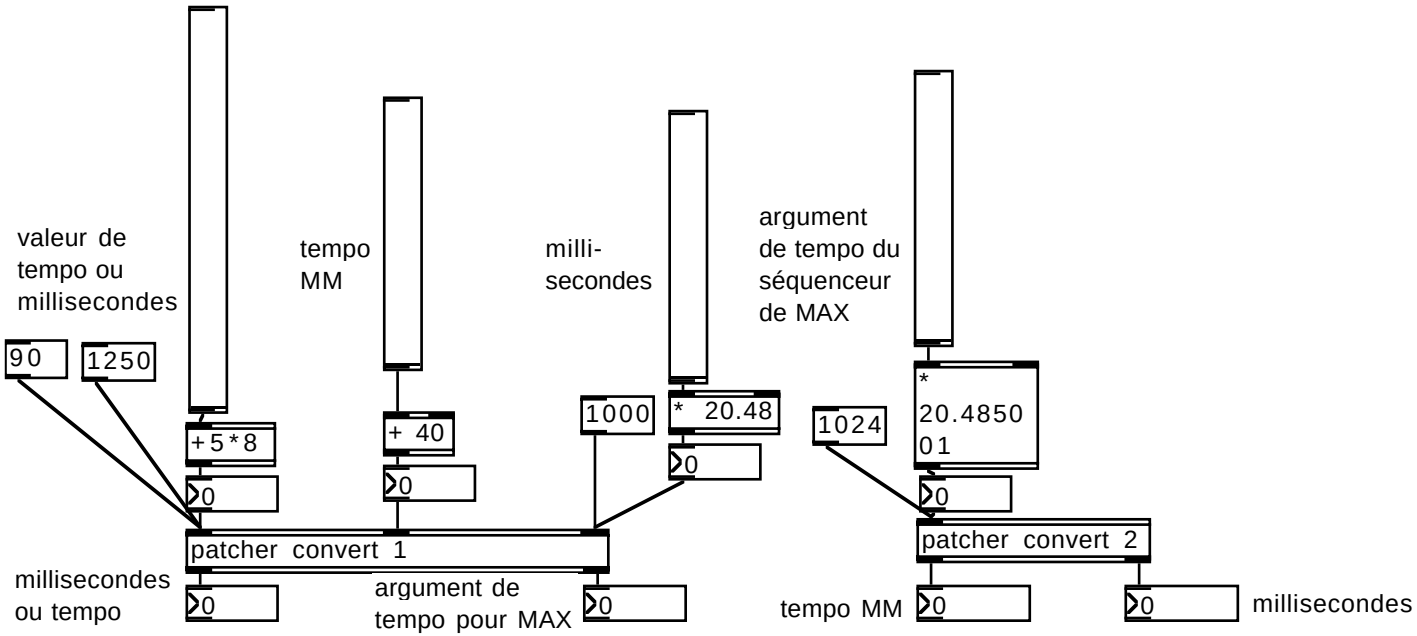
exemple 5d



Mesure du tempo

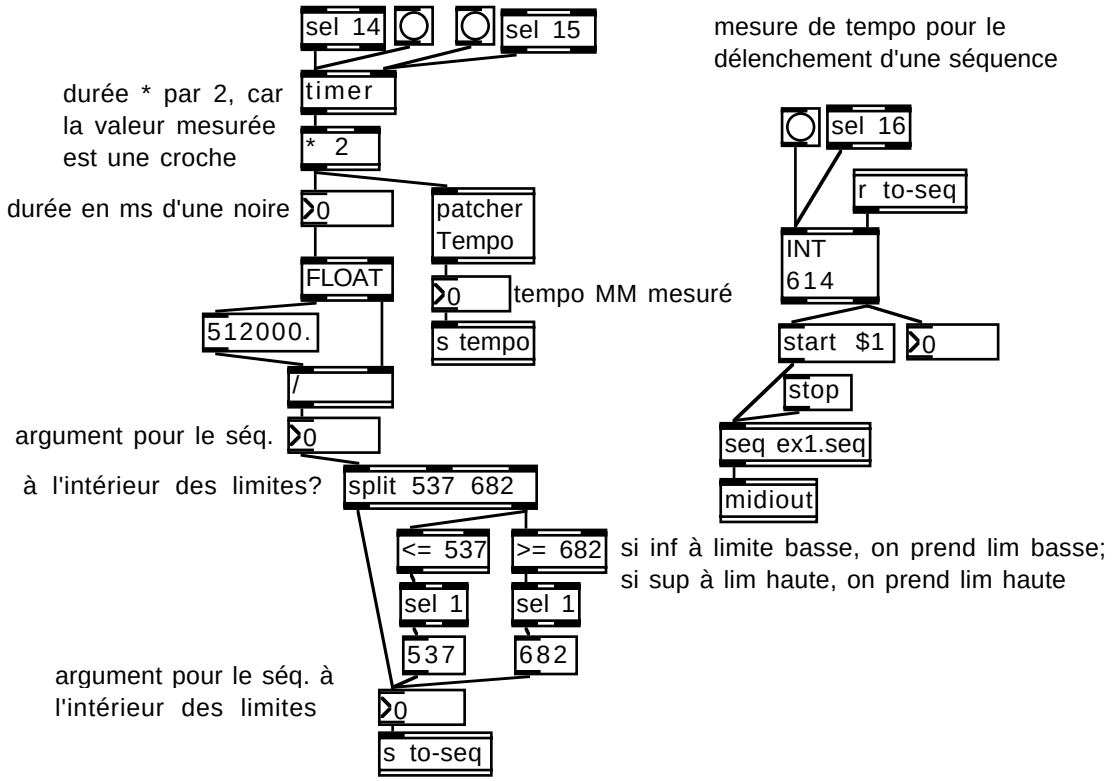
Avant d'aborder un cas concret, il convient de distinguer trois notions liées au temps : 1) la durée (mesurée en millisecondes dans Max) ; 2) le tempo métronomique (MM) ; 3) l'argument de tempo à fournir au séquenceur de Max, sachant qu'il lira Fichier MIDItout fichier MIDI , quelque soit le tempo auquel la Temposéquence a été créée ou sauvegardée dans un autre séquenceur, à 120 à la noire par défaut. Cela correspond à un argument de 1024, tandis que 60 à la noire correspond à un argument de 512. Nous avons donc créé un patcher, l'exemple 6, appelé `Conv Temp`, qui permet le passage entre ces trois grandeurs.

exemple 6



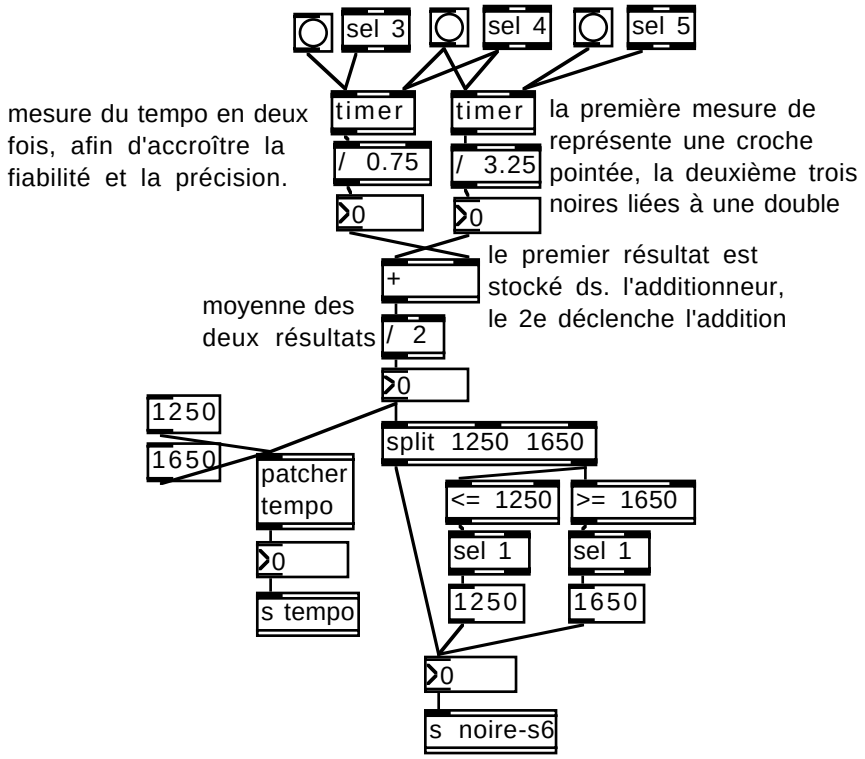
L'exemple 7 illustre la façon retenue pour mesurer le tempo. La durée mesurée doit être multiplié par un facteur la convertissant à la valeur d'une noire. Elle est ensuite convertie en tempo MM et envoyée sur l'écran de commande à fin de visualisation. Parallèlement, elle est convertie en argument de tempo pour le séquenceur de Max, puis testée pour savoir si elle est dans des limites acceptables. Sinon, on lui substitue la valeur limite la plus proche.

exemple 7 (section 1, événements 15 à 17)



L'exemple suivant montre un cas où il a été préférable de prendre le tempo deux fois, car celui-ci étant très lent et les valeurs mesurables n'étant jamais sur les temps, la fiabilité d'une mesure unique n'aurait pas été suffisante. Il s'agit donc de deux objets `timer`, chaque résultat étant converti en la valeur d'une noire, puis leur moyenne calculée. Noter le croisement des connections entre les sorties des `timer` et les entrées de l'additionneur. Ce croisement est nécessaire afin que l'objet + stocke la première valeur, puis ensuite seulement reçoive la deuxième valeur, qui provoque leur addition et l'envoi du résultat à son `outlet`.

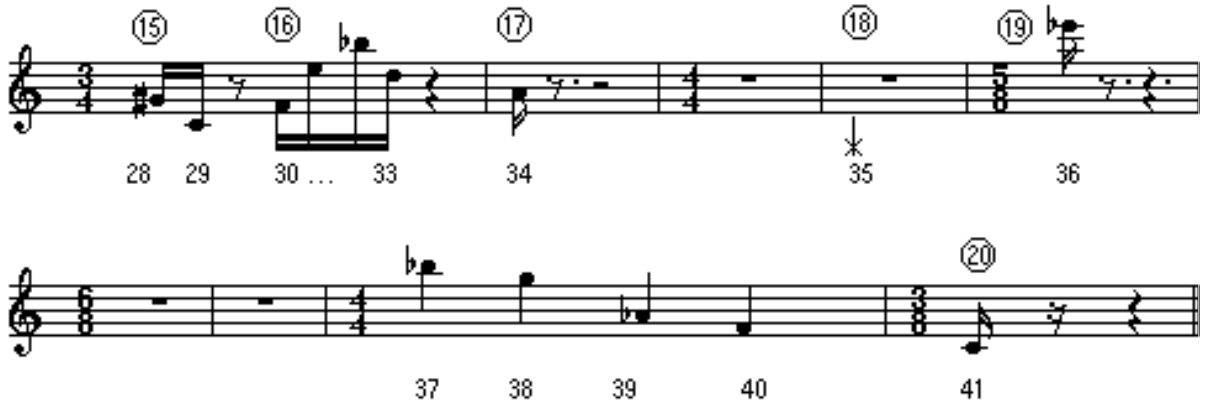
exemple 8 (section 6, événements 3 à 5)



Dans ce cas particulier, la moyenne a été effectuée en donnant un poids égal aux deux valeurs, mais il y a d'autres passages dans la pièce (section 10 notamment) où des mesures successives sont moyennées en donnant un poids plus fort aux mesures les plus récentes, du fait de l'évolution possible du tempo.

Il faut mesurer le tempo le moins longtemps possible avant le départ d'une séquence, celui-ci étant susceptible de fluctuer plus que l'on ne pourrait le croire. Si une séquence dépasse 7 à 10 secondes, il vaut bien mieux la décomposer en plusieurs morceaux, déclenchés chacun à son tour par un événement placé au bon endroit, ou avec un délai si aucune possibilité n'existe à ce moment précis. Cela offre le double avantage de rattraper les décalages (on peut même faire que l'événement qui déclenche une séquence arrête la précédente au cas où celle-ci ne serait pas finie), et d'autoriser un réajustement en cours de route du tempo, chaque nouvelle séquence pouvant démarrer à un tempo corrigé en fonction de mesures prises après le démarrage de la séquence précédente.

Voici un exemple d'une séquence, décomposée en trois. Le premier tronçon est déclenché par l'événement 17, après mesure du tempo entre les évts. 15, 16 et 17 lui-même, une moyenne étant réalisée entre les deux résultats. Le deuxième tronçon est déclenché, après un délai, par l'évt. 19, tandis que le 20 envoie la portion finale.

Cet exemple peut être essayé en studio, en suivant la partition ci-dessous (qui s'enchaîne avec la partition précédente). La séquence n'aura pas tous les sons qu'elle devrait, puisqu'elle utilise normalement deux TX802. TX802Pour avancer le suiveur d[9] [10][11][12]irectement sur l'événement 15, appuyer sur la lettre A (ceci ne vaut que pour l'exemple, et ne correspond pas à l'effet réel du bouton A dans la pièce).

A plusieurs reprises, nous avons eu besoin d'un système pour enlever une réverbération Réverbérationtrès rapidement, tout en gardant une sonorité la plus naturelle possible. Nous avons donc réalisé un double shunt dont les fins arrivent ensemble, le premier pour fermer l'entrée du son dans la réverbération, le deuxième pour fermer la sortie de la réverbération elle-même. L'exemple 9 permet de visualiser l'effet obtenu, et il est facile en changeant les arguments des objets `mult` de varier le retard entre l'événement et le début de l'un ou l'autre des shunts, sans changer le moment de leur fin.

---

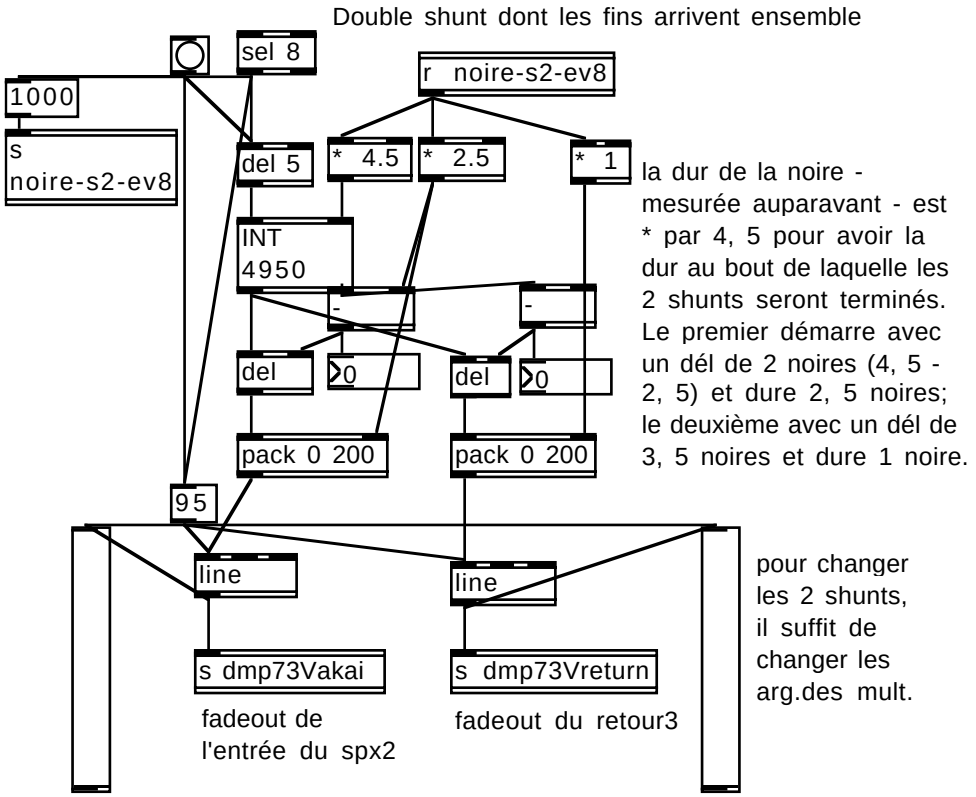
9. Jan Vandenheede avait alors contribué au développement de cet environnement (affichage graphique des hauteurs).

10. Un prédicat est, en Prolog, une relation (et non une fonction) entre des arguments, qui est vraie ou fausse. Il s'agit donc d'une procédure (au sens des langages de programmation) non-déterministe, qui cherche la valeur des arguments respectant la vérité de la relation définie par le prédicat.

11. Le "back-tracking" est ce qui permet à Prolog de chercher successivement toutes les solutions à un problème donné.

12. En Prolog, une variable commence par une lettre suivie de '_'. Ainsi L_p est une variable, mais pas Lp_p.

exemple 9 (section 2, événement 8)


Double shunt dont les fins arrivent ensemble
sel 8
r noire-s2-ev8
1000
s noire-s2-ev8
del 5
* 4.5
* 2.5
* 1
la dur de la noire - mesurée auparavant - est * par 4, 5 pour avoir la dur au bout de laquelle les 2 shunts seront terminés. Le premier démarre avec un dél de 2 noires (4, 5 - 2, 5) et dure 2, 5 noires; le deuxième avec un dél de 3, 5 noires et dure 1 noire.
INT 4950
-
-
del
>0
del
>0
pack 0 200
pack 0 200
95
line
line
pour changer les 2 shunts, il suffit de changer les arg.des mult.
s dmp73Vakai
s dmp73Vreturn
fadeout de l'entrée du spx2
fadeout du retour3

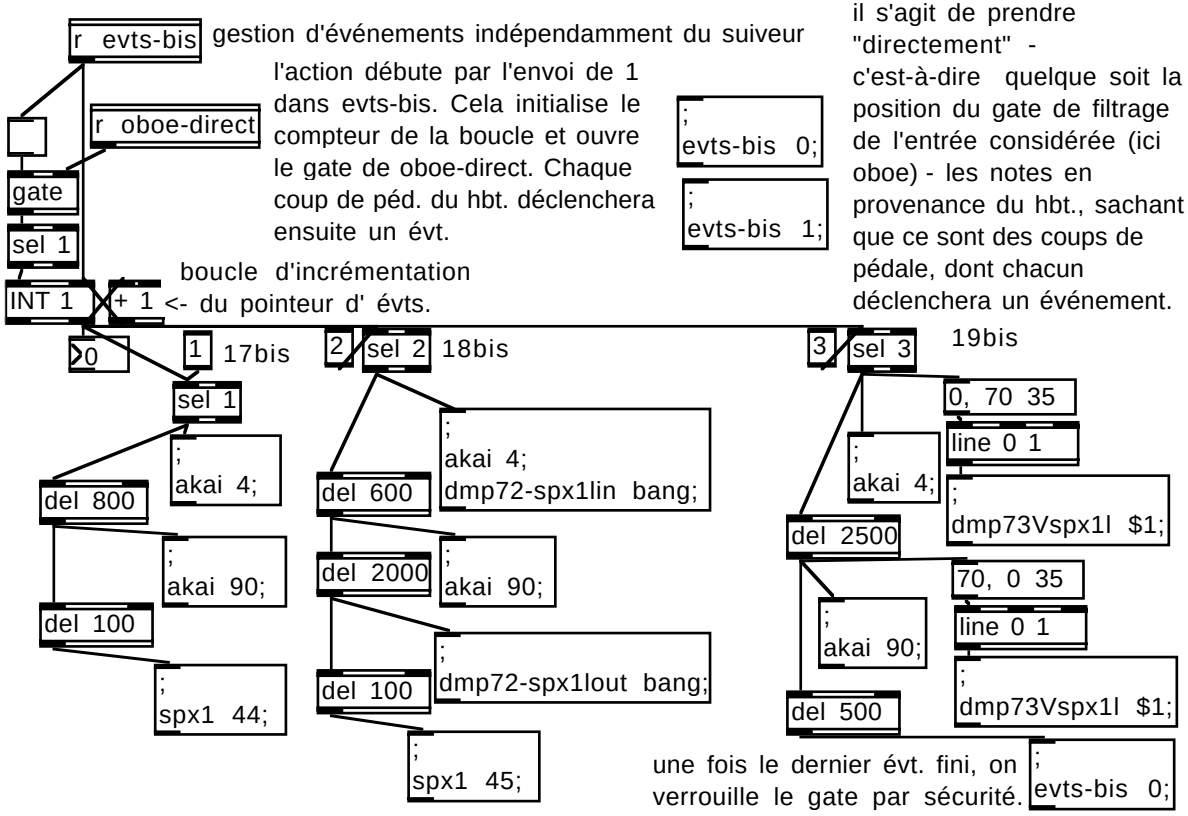
Nous avons ici un passage tutti, avec une séquence de sons synthétisés. Les Clochescloches à vache ainsi que les TX802TX802 rentrent dans l'effet 1 de la DMP7 DMP7en "symphonic", tandis que ces derniers rentrent également dans l'effet 2 en réverbération assez courte. Le Do de Flûtela flûte, gelé dans le PCM, continue jusqu'à l'événement 8, qui comporte le double shunt ci-dessus. La séquence est divisée en deux fragments de deux mesures, et la voix supérieure se prolonge à la fin du premier fragment de façon à parfaire leur enchaînement. Dès que la deuxième partie commence, cette voix, dont le générateur sur le TX802 n'est pas utilisé immédiatement après, est coupée progressivement grâce à l'envoi de *système exclusive*, tel que montré dans l'exemple 4 ci dessus ; voir exemple 4 (section 2, événement 8) on page 37.

L'exemple 10 illustre ce que nous avons appelé les événements bis. Un des passages de l'œuvre demandait à ce que la flûte déclenche un événement quasi-simultanément avec un autre, celui-là déclenché par le hautbois. Il aurait suffit d'une inexactitude rythmique infime de la part de l'un des deux solistes pour que le suiveur soit perdu. Nous avons donc imaginé, plutôt que d'installer un second suiveur, ce qui paraît un peu lourd pour une utilisation si restreinte, un mécanisme permettant de déclencher des événements (appelés "bis") directement par le hautbois. D'où l'intérêt de l'objet `oboe-direct` mentionné plus haut. L'événement précédent déclenche, par l'envoi d'un 1 dans `evts-bis`, l'ouverture du ***gate*** qui auto-

rise l'arrivée des coups de pédale (transformés en note 3) en provenance du hautbois. Chacun de ceux-ci va déclencher l'envoi par l'objet `int`, d'un message, initialement 1, incrémenté après chaque occurence. Le dernier événement se termine en fermant le ***gate*** des `evts-bis`, par sécurité.

---

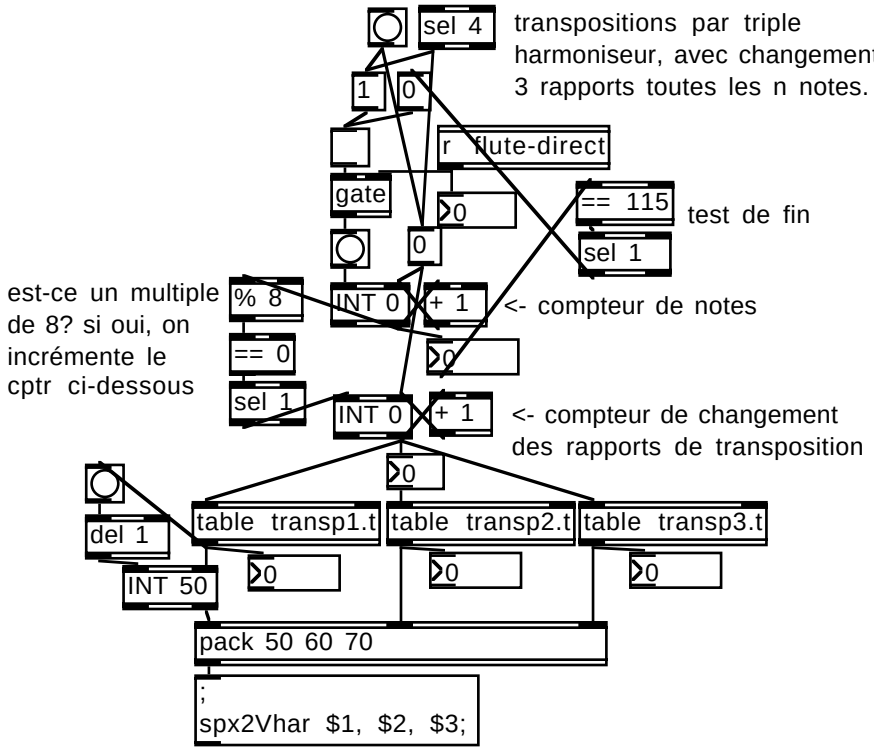
exemple 10 (section 3, événements 17 bis à 19 bis)



C'est ici qu'ont été employés les événements bis (`evts-bis`). Ce passage démontre une alternance rapide de transpositions, une densité très grande d'événements, des transpositions extrêmes (jusqu'à deux octaves et un demi-ton) et multiples.

Dans l'exemple 11 on voit un autre cas d'utilisation des directs. Ici, nous voulons que la Flûteflûte, qui passe par un programme de triple harmoniseur sur la SPX 1000, SPX 1000modifie les rapports de transposition toutes les huit notes qu'elle joue, en fonction de réglages déterminés au préalable. Plutôt que de passer par le suiveur et de faire de chaque envoi de nouveaux rapports de transposition un événement, ce qui en ferait vraisemblablement un nombre beaucoup trop élevé, nous avons préféré recourir à notre `flute-direct`.

exemple 11 (section 9, événement 6)

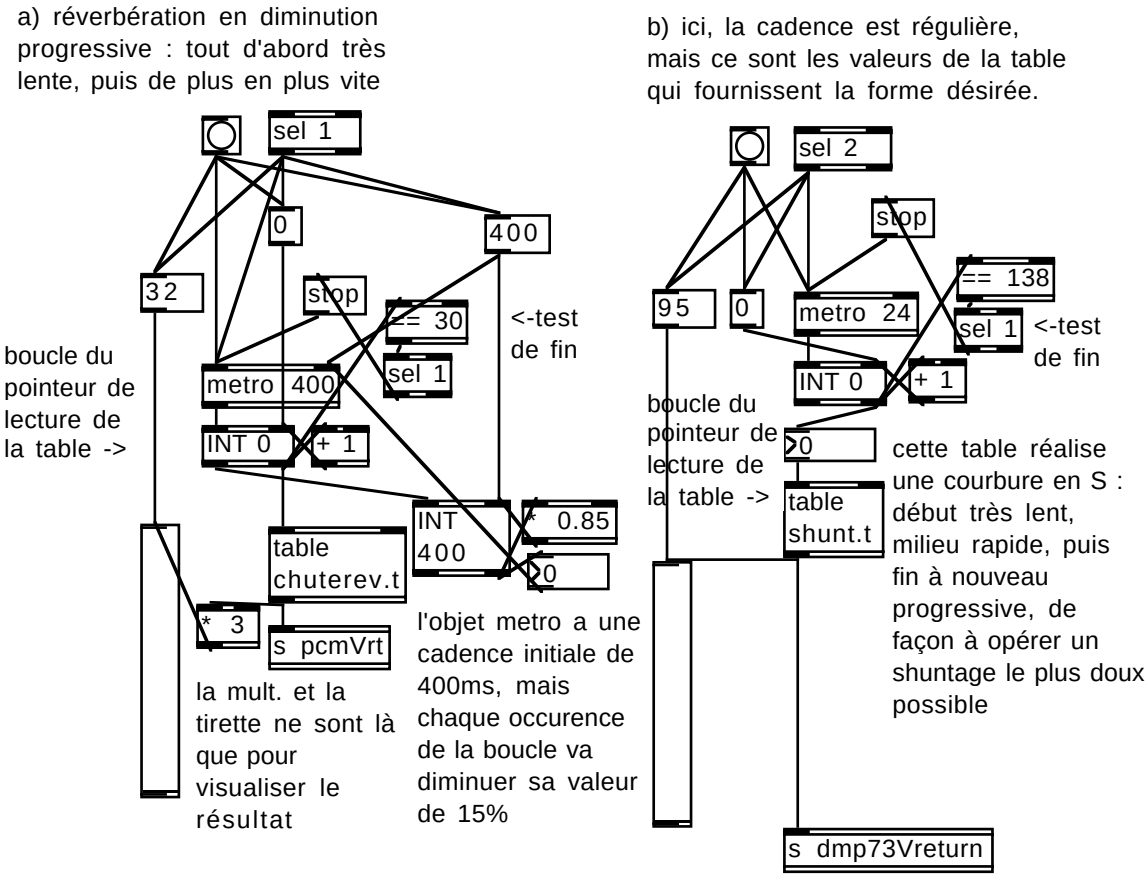


L'événement commence par l'ouverture d'un *gate* qui permet à une première boucle de compter le nombre de notes jouées par la flûteFlûte. Un test permet ensuite d'incrémenter un deuxième compteur chaque fois que le nombre de notes jouées est un multiple de 8 et d'envoyer cette valeur à trois tables contenant les valeurs de transpositions souhaitées.

Dans un premier temps - mm. 194-197 - la flûte est Flûteharmonisée toutes les huit notes de trois rapports de transpositions différents, préalablement rentrés dans trois tables, comme vu ci-dessus. Ensuite - mm. 199-207 - se superpose à cela, de même que rentrent tous les instruments, une séquence à six voix.

Dans Max, l'objet line permet de réaliser très facilement des variations continues. Mais qu'en-est-il si nous voulons des évolutions non-linéaires ? Deux possibilités sont montrées ici.

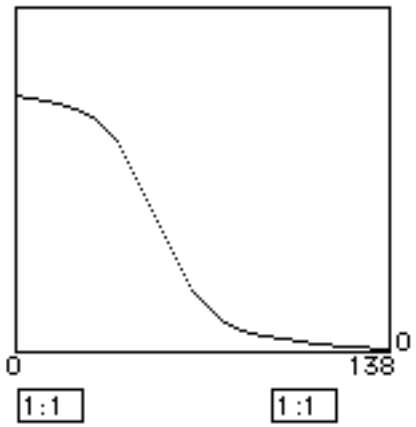
exemple 12 (section 1, événement 6; section 10, événement 1)



Premier événement de l'exemple 12 : diminution assez rapide et progressive de la réverbération sur le programme `Infinite AT` du Lexicon PCM70PCM70. Le problème était d'éviter des clics lors de la diminution progressive, mais la plus rapide possible, du temps de réverbération. Sur cet appareil, on passe directement de la réverbération infinie à une valeur de 32", puis 16", 10", 7,4", etc. Les premières sautes, trop brutales, sont très délicates et demandent à être effectuées doucement, tandis que l'on peut accélérer sans problèmes lorsque l'on s'approche du minimum.

Deuxième événement: shunt très progressif au début, rapide au milieu puis à nouveau progressif à la fin, qui correspond au geste qu'on effectue lorsque l'on veut faire partir un son à la fois rapidement et sans attirer l'attention. En effet, le début et la fin sont les deux moments critiques du point de vue perceptif.

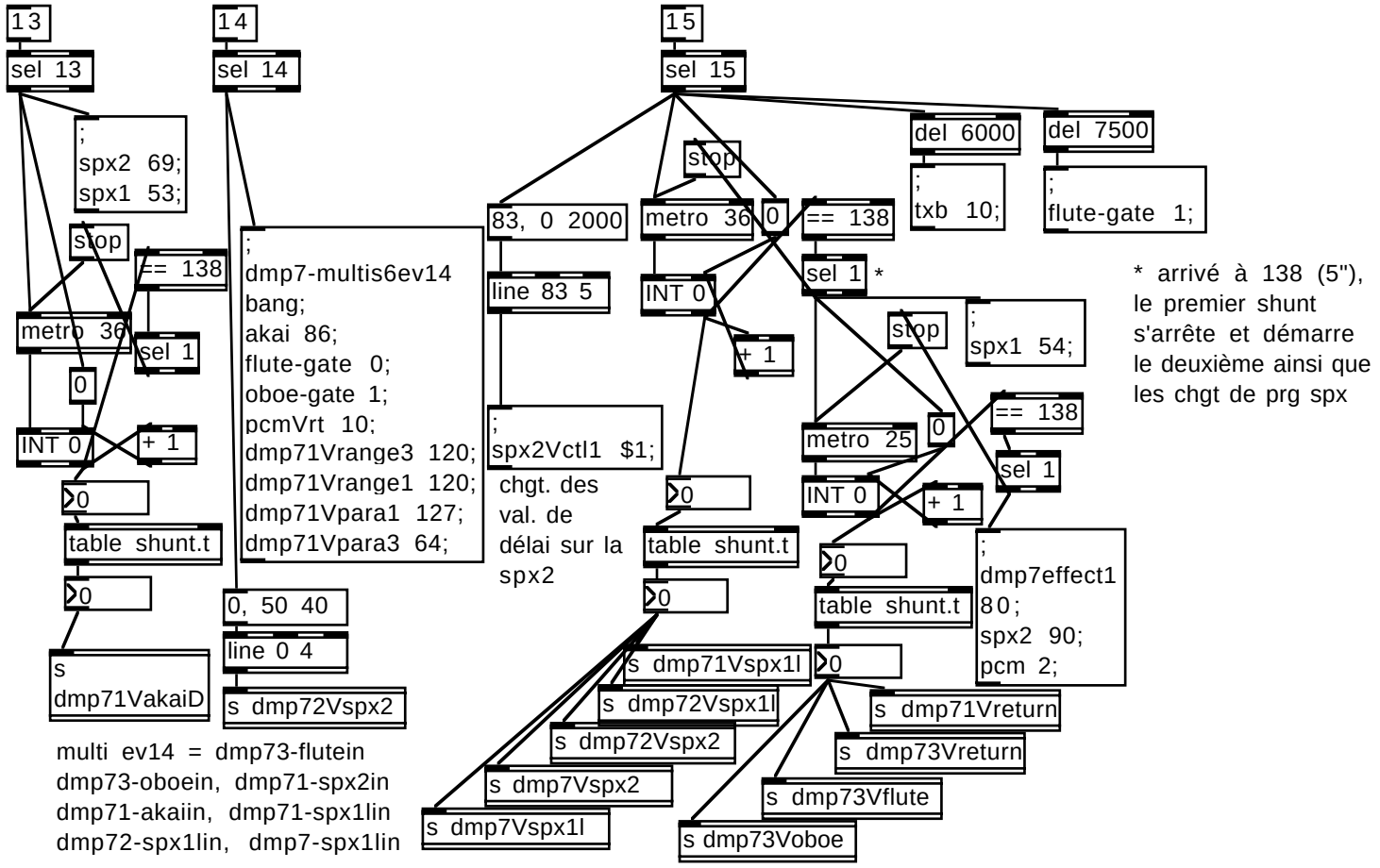
Il est fait appel pour cela à une table, montrée ci-dessous, et qui est lue à cadence régulière.



95 95 94 94 94 94 94 94 93 93
93 93 93 92 92 92 92 91 91 91
90 90 90 89 89 88 88 87 87 86
85 84 83 82 81 80 79 78 76 74
72 70 68 66 64 62 60 58 56 54
52 50 48 46 44 42 40 38 36 34
32 30 28 26 24 22 21 20 19 18
17 16 15 14 13 12 11 10 10 9
9 8 8 7 7 7 6 6 6 6
5 5 5 5 5 5 4 4 4 4
4 4 4 3 3 3 3 3 3 2
2 2 2 2 2 2 2 2 1 1
1 1 1 1 1 1 1 1 1 1
1 1 0 0 0 0 0 0 0 0

Ce passage fait beaucoup appel aux tables shunt, ainsi qu'on peut le voir dans l'exemple suivant.

exemple 13 (section 6, événements 13 à 15)



Dans le milieu de la section 5, l'ensemble rentre dans les effets, puis on a un exemple bien net de sons de synthèse jouée par les claviers.

Un exemple de mode de jeu spécial (Multiphoniques.i.Whistle tones;multiphonique et *whistle tones*) dans des effets, ici des Harmoniseurharmoniseurs en cascades.

D'autres multiphoniques, Multiphoniquescette fois-ci de flûte et de Flûte.i.Hautbois;hautbois dans les Harmoniseurharmoniseurs, tandis que d'autres vents font également des multiphoniques.

Encore un passage avec l'ensemble dans des effets, ici un flanging du PCM70, un peu trop en évidence dans le mixage, auquel s'ajoute une séquence de sons FM.

Remarques diverses - Conclusions

1) L'expérience de cette production a permis de se rendre compte qu'il est toujours préférable, dans le cas d'un tel programme, d'alléger au maximum chaque événement, et notamment de restreindre le plus possible l'utilisation des `del`, quitte à multiplier le nombre d'événements. En effet, les problèmes avec le suiveur sont relativement peu fréquents et assez simples à identifier et à résoudre. En revanche, les problèmes liés à un mauvais ordonnancement temporel des actions MIDI sont, eux, très difficiles à détecter, et toujours dépendants de la circonstance particulière en cours. Ils n'apparaissent souvent que lors des répétitions, en fonction des tempi choisis par les musiciens ou le chef, voire d'éventuelles erreurs de rythme ou de notes, ce qui rend l'identification de la cause du problème très délicate.

2) La mémoire interne des machines MIDI est à utiliser au minimum, voire - quand cela est possible (mais ce n'est le cas, pour cette pièce, que de la DMP7DMP7) - à ne pas utiliser du tout. D'une part cela réduit la dépendance sur ces appareils assez peu fiables dans l'ensemble ; d'autre part, cela permet un meilleur contrôle des états où ils se trouvent à un instant donné. En effet, un problème est toujours plus difficile à identifier s'il s'agit d'une erreur de réglage sur une machine (il faut alors parcourir paramètre après paramètre sur un mauvais panneau d'affichage) que si cela vient de l'ordinateur.

3) Il a été fait une tentative, un peu tard dans la production pour que cela vaille la peine de la pousser au bout, d'afficher sur l'écran tous les paramètres avant leur envoi vers les machines, ce qui permet une lecture aisée de

l'état où se trouve chaque paramètre à tout instant. L'appel très fréquent aux mémoires internes (par changements de programme) a rendu cette tentative peu utile en fin de compte, puisque de nombreuses valeurs affichées étaient en réalité caduques, tandis que beaucoup d'autres (notamment le détail de chaque programme SPXSPX) n'étaient pas affichées.

4) Un des problèmes les plus importants dans ce type de travail est la définition par le compositeur de ce qu'il veut faire, ainsi que sa notation. Il est indispensable, pour mener à bien une telle production, de pouvoir, parfois à plusieurs mois de distance, retrouver dans la partition le résultat attendu, décrit de façon aussi précise que possible. Une quantité considérable de temps se passe en effet à se remémorer des choses précédemment faites et qu'il faut remanier. Pour remédier à cela, ainsi qu'au problème mentionné en 3°, nous avons fait des tableaux d'événements. Ils comportent une ligne pour chaque paramètre contrôlé par le programme. Bien que ne fournissant pas d'indication sur le résultat sonore, ces tableaux ont été précieux pour la mise au point du programme. Il était enfin aisé de repérer la plupart des erreurs, tels que des oublis ou des modifications dont toutes les conséquences n'avaient pas été immédiatement vues. On peut les consulter en annexe à ce texte.

5) A part les problèmes purement techniques et les problèmes de communication, qui font qu'il n'était pas toujours facile de savoir si le résultat désiré était ou non atteint, une des difficultés majeures de ce type de situation est l'équilibre sonore entre les instruments et l'électronique. Inévitable et connue, cette difficulté n'en était pas moins très difficile à maîtriser, car se jouant à plusieurs niveaux. Le volume sonore émis par les instruments est naturellement très fluctuant. Compte tenu de la chaîne particulièrement longue et délicate qui doit être parcourue pour les transformations, il est important d'obtenir en entrée des machines un niveau bien adapté. L'utilisation de limiteurs en sortie des consoles de pré-mixage aide mais ne résout pas entièrement ce problème,

d'autant que les intensités physiques ne sont pas toujours, bien évidemment, corrélées aux niveaux perçus. D'autre part, les transformations subies par les sons affectent ceux-ci de diverses manières, les atténuant ou les renforçant parfois beaucoup. Ensuite, l'amplification influe considérablement sur la perception d'équilibre électronique/instruments. Longue et difficile à mettre en place, elle varie sensiblement d'une répétition à l'autre, surtout quand - cas ordinaire - le dispositif doit être remis en place à chaque fois. Par ailleurs, les sons de synthèse, dont l'équilibrage entre eux est déjà très délicat, sont particulièrement difficiles à mélanger de façon satisfaisante aux sons acoustiques, mêmes amplifiés. Enfin, il est toujours ardu d'identifier, dans tout problème d'équilibre, la part due aux variations dans le jeu des instrumentistes (et à plus forte raison lors des premières répétitions, car les nuances sont traditionnellement leur dernière préoccupation), et la part effectivement due aux machines. Si bien que l'on peut être amené à fortement modifier un réglage, pour le remettre dans une position voisine du départ quelques répétitions plus tard.

L'utilisation de la CAO dans Congruences - par Francis Courtot et Michael Jarrell

Introduction

Ce projet a commencé de façon assez informelle, dans la mesure où il s'agissait d'une recherche compositionnelle en cours, qui a été généralisée pour les besoins de Michael Jarrell. Ce texte n'est pas une description exhaustive des méthodes de composition, mais plutôt un survol de l'utilisation par Michael Jarrell d'un programme. Par ailleurs un nouvel environnement de CAO en PrologII+ est actuellement en cours de réalisation, et ce texte est basé sur ce nouvel environnement, qui contient toutes les fonctionnalités de l'ancien .

Nous décrivons d'abord l'origine du projet, c'est-à-dire ce qui a motivé M. Jarrell à utiliser des programmes d'aide à la composition. Puis nous définissons les caractéristiques employées pour formaliser les objets nécessaires à la composition. Nous décrivons ensuite la manière d'engendrer des objets par contraintes. Enfin, nous examinons l'utilisation effective du programme par le compositeur.

L'origine du projet

Michael Jarrell souhaitait un programme dont le but était de calculer des transformations d'une mélodie initiale, divisée en un certain nombre de groupes. Le but était de générer un nombre défini de hauteurs, dont les intervalles correspondent à ceux des groupes et en respectant les règles suivantes:

- il est possible de permuter les groupes dans n'importe quel ordre.

- pour chaque groupe, les quatre formes usuelles (droite, renversée, contraire, miroir) sont autorisées, à moins que l'on ait précisé des contraintes sur ces formes.

- le nombre de hauteurs provenant de chaque groupe est déterminé par un pourcentage attribué à chaque groupe.

Un exemple permettra de mieux saisir le comportement souhaité du programme.

Soit la mélodie ci-dessous, divisée en 3 groupes:

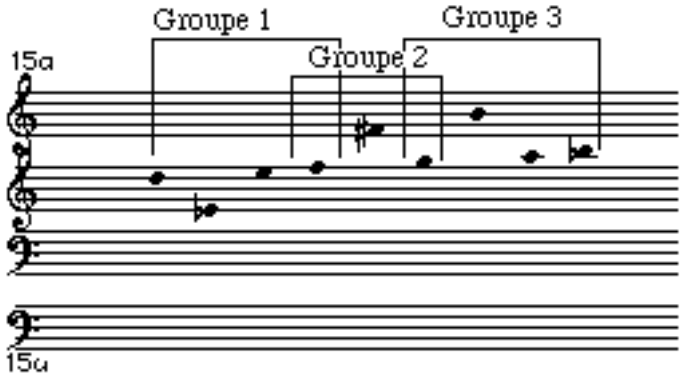


Figure 1: La mélodie initiale.

Et voici la première solution générée par le programme, pour 21 hauteurs engendrées, lorsque l'on donne comme pourcentages respectivement pour chaque groupe: 30 40 30.

```
groupe(
<re.5,mi_bémol.4,mi.5,<fa.5>,fa_dièse.6,<sol.5>,si.6,la.5,si_bémol.5>,
        21,r,constant(droite.d.d'.nil), notes(30.40.30.nil),12)
draw_score(2,r)
fail;

Nombre de répétition(s) de chaque groupe:2.4.2.nil
```

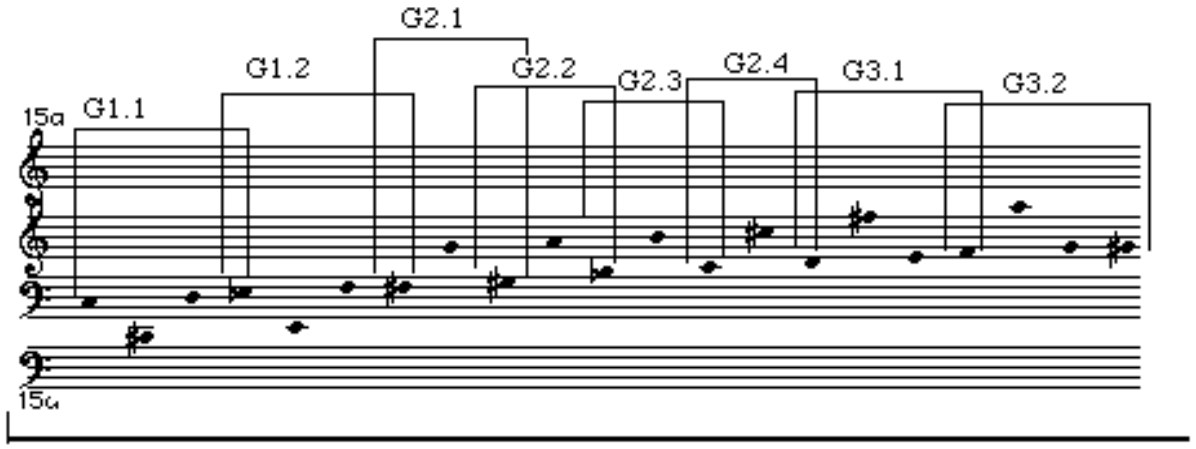


Figure 2: Appel du programme et première solution.

Les valeurs de pourcentage et de nombre de hauteurs engendrées ont conduit le programme à adopter respectivement pour chaque groupe les nombres de répétition 2, 4 et 2. La figure montre les groupes, qui, dans la première solution sont pris dans l'ordre de leur apparition dans la figure initiale.

Voici deux autres solutions générées par le programme.

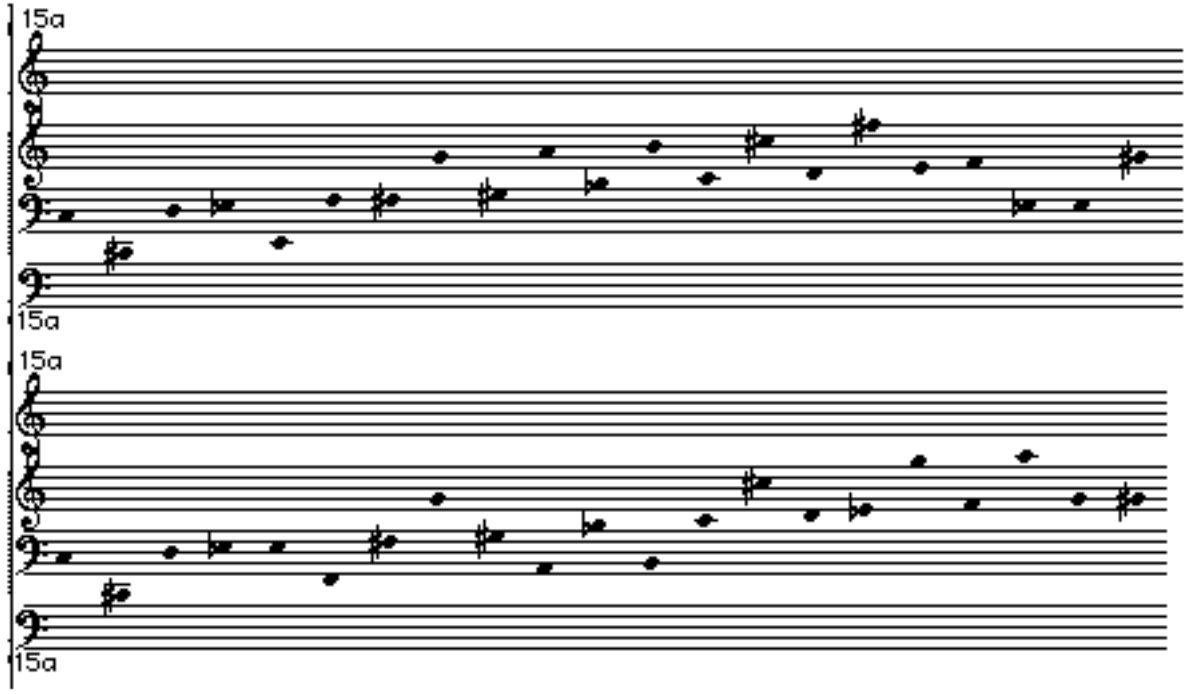


Figure 3: Deux autres solutions.

En haut de la figure 1, figure l'appel du prédicat correspondant au programme souhaité. Celui-ci fixe les groupes initiaux (entre < et >), le nombre de hauteurs souhaitées (21), les contraintes sur les renversement de chaque groupe, les pourcentages (30, 40, 30) et le modulo à l'intérieur duquel les intervalles seront considérés (12 pour le modulo d'octave). Par retour-arrière ("back-tracking"), ce prédicat va engendrer toutes les possibilités définies par les règles précédentes.

Les solutions des figures 2 et 3 montrent rapidement comment la déformation de la mélodie initiale respecte les données intervalliques prises dans chaque groupe. C'est précisément cette propriété de dérivation qui intéressait M. Jarrell dans l'expérimentation (la notion de responsabilité intervallique, d'une solution à une autre). En effet chaque solution devait dans la partition finale être superposée, la figure initiale provenant d'une phrase d'un des instruments solistes. La seconde solution proposée ci-dessous est assez éloignée de la première (plusieurs centaines de solutions, compte tenu des valeurs, sont nécessaires avant d'arriver à celle-ci).

Associées à ce mécanisme d'engendrement, nous avons défini certaines contraintes qui permettent de réduire le nombre de solution. Ces contraintes s'appuient sur des harmonies, provenant d'un autre processus d'engendrement (verticalisation de phrases des solistes), et correspondent à un "gel" variable des hauteurs.

Le problème auquel nous avons été confronté était le nombre de possibilités ainsi définies (pour un appel du prédicat ci-dessus correspondant à Congruences, et sans contraintes, onze milliards de possibilités, d'où à l'époque environ trois siècles de calcul!), et aussi la proximité entre un résultat et un autre.

Pour ces raisons, et en étudiant de plus près les premiers résultats obtenus, nous avons décidé de nous orienter vers la réutilisation d'un de mes prédicats. Celui-ci engendre un certain nombres de valeurs consécutives, musicalement utilisables, que nous appelons forme intervallique. Si l'utilisateur décide d'associer des hauteurs à ces valeurs, on obtient de la sorte une catégorisation et un engendrement de "mélodies". L'utilisation conjointe de ce programme et de contraintes permet de définir plus précisément le "style" des solutions obtenues.

Méthodologie et mise en œuvre

Catégorisation.

L'idée sous-jacente aux prédicats que nous allons voir est que le contenu intervallique est une caractéristique de base d'une "mélodies". Ce contenu intervallique est lui-même considéré de la façon la plus abstraite possible, c'est à dire comme une suite de relations (intervalles) d'un élément à son successeur, dont la définition peut être variable (il y a plusieurs instances possibles pour une même suite d'intervalles), à cause de la notion de modulo. Nous décrivons une forme intervallique par 8 caractéristiques. Ces caractéristiques sont toutes structurelles et non définies dans l'intention de connaître des attributs perceptifs attachés aux objets musicaux. Nous définissons d'abord un *nom* à la suite d'intervalles abstraite choisie. Un prédicat permet d'associer un nom à une suite arbitraire d'intervalle. Nous caractérisons ensuite une *permutation* de cette suite. Nous nous sommes limité aux seules permutations circulaires, afin de conserver une certaine identité à la suite d'intervalles.

Dans la caractéristique appelée *renversement*, nous trouvons les quatre possibilités usuelles: droite, renversée, contraire, miroir. Le *profil* de la suite de valeurs est caractérisé comme une succession de *Plus* et de *Moins*, qui correspond au sens des intervalles liant chaque valeur successive.

La caractéristique suivante est le *modulo* utilisé pour la suite de valeurs. Pour une suite de hauteurs, ce modulo est couramment l'octave (12), mais on peut en utiliser d'autres (il en existe un nombre défini théoriquement en fonction du nombre d’intervalle).

La caractéristique suivante concerne l'intervalle *total* obtenu entre la première et la dernière valeur (somme de tous les intervalles associé à leur signe - *Plus* ou *Moins*). Cette valeur est définie comme un doublet, dont la première valeur est l'intervalle calculé compte tenu du modulo, tandis que la seconde est l'intervalle réel.

L'avant dernière caractéristique est la *forme intervallique* (liste de valeurs).

La dernière concerne les répétitions que l'on trouve dans la suite de valeurs. Cette caractéristique appelée *symétrie*. Ces répétitions sont définies compte tenu du modulo.

Un exemple permet de mieux saisir ces définitions:

Nom: bb;

Permutation: ton, tierce_mineure, septième_majeure, tierce_mineure, septième_majeure, tierce_mineure, septième_majeure, sixte_mineure;

Renversement: droit;

Profil: Plus, Plus, Plus, Plus, Plus, Plus, Plus, Plus;

Modulo: octave;

Total: <52, 4>

Forme intervallique: do3, re3, fa3, mi4, sol4, fa#5, la5, sol#6, mi7.

Symétrie: 4-9;

Voici, en Prolog, comment cet exemple est écrit:

figure_hauteur(

bb,ton.tierce_mineure.septième_majeure.tierce_mineure.

septième_majeure.tierce_mineure.septième_majeure.sixte_mineure.nil,droite,

Plus.Plus.Plus.Plus.Plus.Plus.Plus.Plus.nil,12,<52, 4>,

(do.3).(re.3).(fa.3).(mi.4).(sol.4).(fa_dièse.5).(la.5).

(sol_dièse.6).(mi.7).nil,(4.9.nil).nil);

Engendrement

Le même prédicat permet d'engendrer des formes intervalliques dont on connaît tout ou partie (voire aucune) des caractéristiques précédentes. Ce prédicat permet en fait de chercher à la fois les valeurs numériques et les hauteurs correspondantes à ces valeurs. De cette façon, le programme permet de chercher toutes les formes intervalliques correspondantes aux valeurs données comme argument au prédicat. Par exemple, pour chercher toutes les formes intervalliques dont le nom est bb, le modulo 12 et le total, compte tenu du modulo, une tierce majeure, il faut appeler le prédicat suivant:

figure_hauteur(bb,P_er,R_env,A_ll,12,<tierce_majeure,T_ot>,

(do.3).L_haut,S_ym);

Voici les premiers résultats, sous forme de hauteurs:

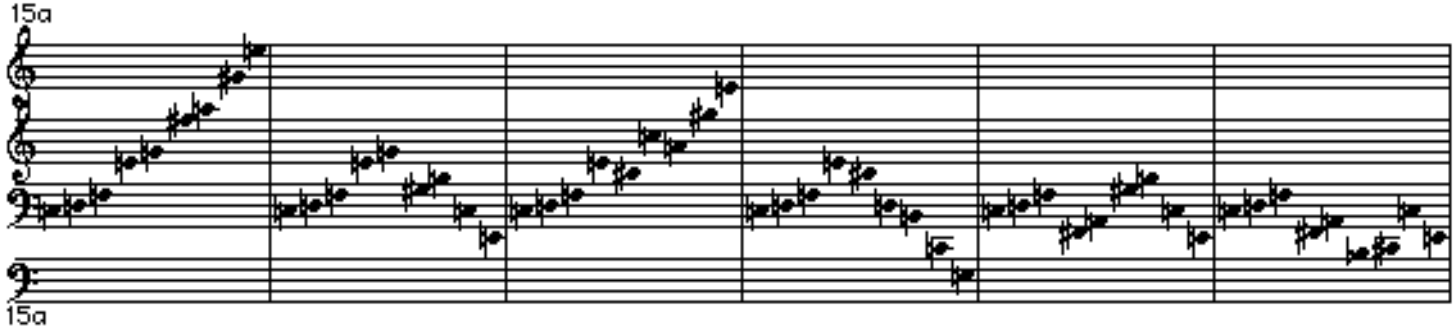



Figure 4: Quelques formes intervalliques.

Plus formellement, nous pouvons regrouper en quatre types les prédicats qui sont impliqués dans l'engendrement:

* Génération: dans le cas que nous décrivons, c'est l'un des prédicats qui se chargent de calculer des Forme intervalliques.

* Association : ces prédicats permettent d'associer des valeurs d'un type différent à une forme intervallique (ou le contraire). Par exemple, le prédicat `jolie_hauteur (X,Y)`, dans lequel `X` est une liste d'entiers et `Y` une liste de hauteurs, se charge de transformer une liste d'entiers en une liste de hauteurs (ou vice versa), selon celui qui est connu (nous ne sommes pas dans un contexte de programmation fonctionnelle mais logique, relationnelle).

* Prédicats de contraintes : ces prédicats permettent d'assurer la présence ou l'absence d'une valeur (dans une position quelconque ou connue) dans la forme intervallique, d'assurer un "gel" (variable) des registres dans une liste de hauteurs associés, de donner des valeurs extrêmes aux valeurs de la forme intervallique, etc…

* Présentation des résultats : par exemple, le prédicat `draw_score` qui permet d'afficher sous forme de partition les résultats de l'engendrement.

D'autres prédicats existent, correspondant à la manipulation des objets (par exemple liste de hauteurs). Ceux-ci n'ont finalement pas été utilisés dans la production de cette œuvre. En suggérant à Michael Jarrell d'utiliser le programme décrit plus haut pour contrôler la dérivation d'une "mélodie", nous perdions l'aspect permutation des groupes, mais gagnions en clarté de définition des résultats engendrés.

L'utilisation du programme par Michael Jarrell.

Michael Jarrell souhaitait superposer aux solistes de la pièce des "mélodies" dérivées de celles des solistes. Pour cela, il était nécessaire d'obtenir un nombre assez important de solutions, pour permettre à presque tout l'orchestre de participer à cet "agrandissement" des solistes (ce processus d'agrandissement a d'ailleurs une valeur formelle importante, sous une forme ou une autre pour toute la pièce).

Notes du programme de la création

Congruences est ma première expérience avec électronique. Tout en employant des systèmes de type personnels (Macintosch et instruments MIDI), j'ai cherché à réaliser l'équivalent de certains effets obtenus à partir de moyens plus importants. Cette démarche sous-entend un grand nombre de commandes et d'actions MIDI (entre autres pour masquer des faiblesses évidentes de ces machines).

Pour ce faire, le programme le plus adapté était incontestablement Patcher (Max) récemment développé à l'Ircam par Miller Puckette.

Les contrôles installés pour les éléments électroniques de la pièce doivent être très flexibles et certains aspects d'interprétation sont assurés par ordinateur. Ce dernier traite donc la partition, organise les matériaux d'après les ordres donnés par les musiciens et gère les instruments MIDI, jouant aussi des matériaux prédéfinis combinés en temps réel. Par exemple, quand le musicien donne "une battue", l'ordinateur vérifie le temps courant de manière à ralentir ou à accéléer le ou les éléments. En ce qui concerne l'écriture musicale de la pièce, la quintessence de *Congruences* est tirée de quelques mesures d'une oeuvre antérieure *Trace-Ecart.* Une section a été réalisée avec l'aide d'un programme de Composition Assistée par Ordinateur (CAO) développé en Prolog par Francis Courtot à l'Ircam ; je tiens à le remercier ainsi que Miller Puckette, mais je voudrais surtout exprimer ma reconnaissance à Jan Vandenheede et Nicolas Vérin pour leur patience et leur aide sur les nombreux aspects techniques de la pièce.

Michaël Jarrell